aws

Guide de l'utilisateur

Amazon Simple Storage Service

Version de l'API 2006-03-01

Amazon Simple Storage Service: Guide de l'utilisateur

Table of Contents

..... mmmdcxc

Qu'est-ce qu'Amazon S3 ?

Amazon Simple Storage Service (Amazon S3) est un service de stockage d'objets qui offre une évolutivité, une disponibilité des données, une sécurité et des performances de pointe. Les clients de toutes tailles et secteurs peuvent utiliser Amazon S3 pour stocker et protéger toute quantité de données dans un large éventail de cas d'utilisation, par exemple des lacs de données, des sites Web, des applications mobiles, des sauvegardes et restaurations, des archives, des applications métier, des appareils IoT et des analyses de Big Data. Amazon S3 offre des fonctions de gestion qui vous permettent d'optimiser, d'organiser et de configurer l'accès à vos données aux fins de répondre aux exigences spécifiques de votre entreprise, de votre organisation et de votre conformité.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Fonctions d'Amazon S3

Classes de stockage

Amazon S3 offre un large éventail de classes de stockage conçues pour différents cas d'utilisation. Par exemple, vous pouvez stocker les données de production stratégiques dans S3 Standard ou S3 Express One Zone pour un accès fréquent, réduire les coûts en stockant les données rarement

consultées dans S3 Standard-IA ou S3 One Zone-IA, et archiver les données à des coûts minimaux dans S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.

Amazon S3 Express One Zone est une classe de stockage Amazon S3 à zone unique et hautes performances, spécialement conçue pour fournir un accès aux données constant en moins de dix millisecondes pour vos applications les plus sensibles à la latence. S3 Express One Zone est la classe de stockage d'objets cloud à latence la plus faible disponible à ce jour, avec des vitesses d'accès aux données jusqu'à 10 fois plus rapides et des coûts de demande 50 % inférieurs à ceux de S3 Standard. S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d'objets et les ressources de calcul, ce qui assure la vitesse d'accès la plus élevée possible. En outre, pour augmenter encore la vitesse d'accès et prendre en charge des centaines de milliers de demandes par seconde, les données sont stockées dans un nouveau type de compartiment : un compartiment de répertoires Amazon S3. Pour plus d'informations, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Vous pouvez stocker des données avec des modèles d'accès changeants ou inconnus dans S3 Intelligent-Tiering, ce qui optimise les coûts de stockage en déplaçant automatiquement vos données entre quatre niveaux d'accès lorsque vos modèles d'accès changent. Ces quatre niveaux d'accès comprennent deux niveaux d'accès à faible latence optimisés pour les accès fréquents et peu fréquents, et deux niveaux d'accès d'archive optionnels conçus pour un accès asynchrone pour les données rarement consultées.

Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Gestion du stockage

Amazon S3 dispose de fonctions de gestion du stockage que vous pouvez utiliser pour gérer les coûts, répondre aux exigences réglementaires, réduire la latence et enregistrer plusieurs copies distinctes de vos données aux fins de respecter les exigences de conformité.

- Cycle de vie S3 : configurez une configuration de cycle de vie pour gérer vos objets et les stocker de manière rentable pendant tout leur cycle de vie. Vous pouvez transférer des objets vers d'autres classes de stockage S3 ou faire expirer des objets qui atteignent la fin de leur durée de vie.
- S3 Object Lock – Empêchez que les objets Amazon S3 soient supprimés ou écrasés sur une période déterminée ou indéfinie. Vous pouvez utiliser Object Lock pour répondre aux exigences réglementaires qui nécessitent un stockage write-once-read-many(WORM) ou simplement pour

ajouter un niveau de protection supplémentaire contre les modifications et les suppressions d'objets.

- Réplication S3 : répliquez les objets ainsi que leurs métadonnées et balises d'objet respectives vers un ou plusieurs compartiments de destination identiques ou différents Régions AWS pour réduire la latence, la conformité, la sécurité et d'autres cas d'utilisation.
- Opérations par lots S3 – Gérez des milliards d'objets à grande échelle avec une seule demande API S3 ou quelques clics dans la console Amazon S3. Vous pouvez utiliser Batch Operations pour effectuer des opérations telles que Copy, Invoke AWS Lambda function et Restore sur des millions ou des milliards d'objets.

Gestion des accès et sécurité

Amazon S3 fournit des fonctions d'audit et de gestion de l'accès à vos compartiments et objets. Par défaut, les compartiments S3 et les objets sont privés. Vous avez uniquement accès aux ressources S3 que vous créez. Pour accorder des autorisations de ressources granulaires prenant en charge votre cas d'utilisation spécifique ou pour vérifier les autorisations de vos ressources Amazon S3, vous pouvez utiliser les fonctions suivantes.

- Bloquer l'accès public S3 – Bloque l'accès public aux compartiments S3 et aux objets. Par défaut, les paramètres de blocage de l'accès public sont activés au niveau du compartiment. Nous vous recommandons de laisser tous les paramètres de blocage de l'accès public activés, sauf si vous savez que vous devez en désactiver un ou plusieurs pour votre cas d'utilisation spécifique. Pour plus d'informations, consultez Configuration des paramètres de blocage d'accès public pour vos compartiments S3.
- AWS Identity and Access Management (IAM) — IAM est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources, y compris à vos ressources Amazon S3. Avec IAM, vous pouvez gérer de manière centralisée les autorisations qui contrôlent les AWS ressources auxquelles les utilisateurs peuvent accéder. Vous pouvez utiliser IAM pour contrôler les personnes qui s'authentifient (sont connectées) et sont autorisées (disposent d'autorisations) à utiliser des ressources.
- Politiques de compartiment – Utilisez un langage de politique basé sur IAM pour configurer les autorisations basées sur les ressources de vos compartiments S3 et des objets qu'ils contiennent.
- Amazon S3 access points (Points d'accès Amazon S3) — configurez des points de terminaison réseau nommés avec des stratégies d'accès dédiées pour gérer l'accès aux données à grande échelle pour les jeux de données partagés dans Amazon S3.

- Listes de contrôle d'accès (ACL) – Accordez des autorisations de lecture et d'écriture pour des compartiments et des objets individuels aux utilisateurs autorisés. En règle générale, nous recommandons d'utiliser des politiques basées sur les ressources S3 (politiques de compartiment et politiques de point d'accès) ou des politiques d'utilisateur IAM pour le contrôle d'accès à la place des listes ACL. Les politiques constituent une option de contrôle d'accès simplifiée et plus flexible. Les politiques de compartiment et de point d'accès vous permettent de définir des règles s'appliquant de manière générale à toutes les demandes adressées à vos ressources Amazon S3. Pour plus d'informations sur les cas spécifiques où vous devriez utiliser des listes ACL plutôt que des politiques basées sur les ressources ou des politiques d'utilisateur IAM, consultez Gestion des accès à l'aide des listes ACL.
- Propriété d'objets S3 : prenez possession de chaque objet présent dans votre compartiment, afin de simplifier la gestion des accès aux données stockées dans Amazon S3. La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour désactiver ou activer les listes ACL. Par défaut, les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets présents dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès.
- Analyseur d'accès IAM pour S3 : évaluez et contrôlez vos politiques d'accès aux compartiments S3, en veillant à ce que ces politiques fournissent uniquement l'accès prévu à vos ressources S3.

Traitement des données

Pour transformer les données et déclencher des flux de travail afin d'automatiser diverses autres activités de traitement à grande échelle, vous pouvez utiliser les fonctions suivantes.

- S3 Object Lambda – ajoutez votre propre code aux requêtes S3 GET, HEAD et LIST afin de modifier et de traiter les données lorsqu'elles sont renvoyées vers une application. Filtrez les lignes, redimensionnez dynamiquement les images, supprimez les données confidentielles et bien plus encore.
- Notifications d'événements : déclenchez des flux de travail qui utilisent Amazon Simple Notification Service (Amazon SNS), Amazon Simple Queue Service (Amazon SQS) et lorsqu'une modification AWS Lambda est apportée à vos ressources S3.

Journalisation et surveillance du stockage

Amazon S3 fournit des outils de journalisation et de surveillance que vous pouvez utiliser pour surveiller et contrôler la façon dont vos ressources Amazon S3 sont utilisées. Pour plus d'informations, consultez Outils de surveillance.

Outils de surveillance automatique

- Amazon CloudWatch Metrics for Amazon S3 — Suivez l'état de fonctionnement de vos ressources S3 et configurez des alertes de facturation lorsque les frais estimés atteignent un seuil défini par l'utilisateur.
- AWS CloudTrail— Enregistrez les actions effectuées par un utilisateur, un rôle ou un Service AWS dans Amazon S3. CloudTrail les journaux vous fournissent un suivi détaillé des API pour les opérations au niveau du bucket S3 et au niveau de l'objet.

Outils de surveillance manuelle

- La journalisation des accès au serveur fournit des enregistrements détaillés pour les demandes soumises à un compartiment. Vous pouvez utiliser les journaux d'accès au serveur pour des audits de sécurité et d'accès, afin d'en savoir plus sur votre base de clients et pour comprendre votre facture Amazon S3.
- AWS Trusted Advisor — Évaluez votre compte à l'aide de vérifications des AWS meilleures pratiques pour identifier les moyens d'optimiser votre AWS infrastructure, d'améliorer la sécurité et les performances, de réduire les coûts et de surveiller les quotas de service. Vous pouvez ensuite suivre les recommandations pour optimiser vos services et vos ressources.

Analytique et informations

Amazon S3 offre des fonctions qui vous aident à gagner en visibilité sur l'utilisation de votre stockage, ce qui vous permet de mieux comprendre, analyser et optimiser votre stockage à grande échelle.

- Amazon S3 Storage Lens : comprenez, analysez et optimisez votre stockage. S3 Storage Lens fournit plus de 60 indicateurs d'utilisation et d'activité ainsi que des tableaux de bord interactifs pour agréger les données relatives à l'ensemble de votre organisation, à des comptes Régions AWS, à des compartiments ou à des préfixes spécifiques.
- Analyse de classe de stockage : analysez les modèles d'accès au stockage afin de décider lorsqu'il est temps de déplacer vos données vers une classe de stockage plus économique.

- Inventaire S3 avec rapports d'inventaire : auditez et produisez des rapports sur les objets et leurs métadonnées correspondantes, et configurez d'autres fonctions Amazon S3 pour qu'elles agissent dans les rapports d'inventaire. Par exemple, vous pouvez signaler le statut de réplication et de chiffrement de vos objets. Pour obtenir la liste de toutes les métadonnées disponibles pour chaque objet dans les rapports d'inventaire, consultez Liste d'inventaire Amazon S3.

Forte cohérence

Amazon S3 assure une forte read-after-write cohérence pour les requêtes PUT et DELETE relatives à l'ensemble des objets de votre compartiment Amazon S3 Régions AWS. Cela s'applique à la fois aux écritures sur de nouveaux objets ainsi qu'aux opérations PUT qui écrasent les objets existants et aux opérations DELETE. En outre, les opérations de lecture sur Amazon S3 Select, les listes de contrôle d'accès Amazon S3, les balises d'objet Amazon S3 et les métadonnées d'objet (par exemple l'objet HEAD) sont fortement cohérentes. Pour plus d’informations, consultez Modèle de cohérence des données Amazon S3.

Fonctionnement d'Amazon S3

Amazon S3 est un service de stockage d'objets qui stocke les données en tant qu'objets dans des compartiments. Un objet est un fichier et toutes les métadonnées qui le décrivent. Un compartiment est un conteneur d'objets.

Pour stocker vos données dans Amazon S3, vous devez d'abord créer un compartiment et spécifier un nom de compartiment et une Région AWS. Ensuite, vous chargez vos données dans ce compartiment en tant qu'objets dans Amazon S3. Chaque objet possède une key (clé) (ou key name [nom de clé]), qui est l'identifiant unique d'un objet au sein d'un compartiment.

S3 propose des fonctionnalités que vous pouvez configurer pour prendre en charge votre cas d'utilisation spécifique. Par exemple, vous pouvez utiliser la gestion des versions S3 pour conserver plusieurs versions d'un objet dans un même compartiment et vous permettre de restaurer des objets qui sont accidentellement supprimés ou écrasés.

Les compartiments et les objets qu'ils contiennent sont privés et ne sont accessibles que si vous accordez explicitement des autorisations d'accès. Vous pouvez utiliser des politiques de compartiment, des politiques AWS Identity and Access Management (IAM), des listes de contrôle d'accès (ACL) et des points d'accès S3 pour gérer l'accès.

Rubriques

- Compartiments
- Objets
- Clés
- Gestion des versions S3
- ID de version
- Politique de compartiment
- Points d'accès S3
- Listes de contrôle d'accès (ACL)
- Régions

Compartiments

Un compartiment est un conteneur d'objets stockés dans Amazon S3. Chaque compartiment permet de stocker un nombre illimité d'objets et vous pouvez avoir jusqu'à 100 compartiments dans votre compte. Pour demander une augmentation, reportez-vous à la console Service Quotas.

Chaque objet est contenu dans un compartiment. Par exemple, si l'objet nommé `photos/puppy.jpg` est stocké dans le compartiment `DOC-EXAMPLE-BUCKET`, dans la région USA Ouest (Oregon), il est adressable à l'aide de l'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Pour plus d'informations, consultez Accès à un compartiment.

Lorsque vous créez un compartiment, vous saisissez un nom de compartiment et choisissez la Région AWS où se trouve le compartiment. Une fois un compartiment créé, vous ne pouvez pas modifier son nom ni sa Région. Les noms de compartiments doivent suivre les règles de dénomination de compartiment. Vous pouvez également configurer un compartiment pour utiliser Gestion des versions S3 ou autre Gestion du stockageFonctions de

Les compartiments aussi :

- Organisez l'espace de noms Amazon S3 au niveau le plus élevé.
- Identifiez le responsable de compte pour les frais de stockage et de transfert de données.
- Fournissez des options de contrôle d'accès, telles que des stratégies de compartiment, des listes de contrôle d'accès (ACL) et des points d'accès S3, que vous pouvez utiliser pour gérer l'accès à vos ressources Amazon S3.
- Sert d'unité de regroupement pour les rapports d'utilisation.

Pour plus d'informations sur les compartiments, consultez Présentation des compartiments.

Objets

Les objets sont les entités fondamentales stockées dans Amazon S3. Les objets sont composés de données et de métadonnées. Les métadonnées sont un ensemble de paires nom-valeur décrivant des objets. Ces pairs comprennent certaines métadonnées par défaut telles que la date de la dernière modification et des métadonnées HTTP standard comme `Content-Type`. Vous pouvez aussi spécifier des métadonnées personnalisées au moment du stockage de l'objet.

Un objet est identifié de manière unique dans un compartiment parclé (nom)et un ID de version (si S3 Versioning est activé sur le compartiment). Pour en savoir plus sur les objets, consultez Présentation des objets Amazon S3.

Clés

Une clé d'objet (ou nom de clé) est l'identifiant unique d'un objet au sein d'un compartiment. Chaque objet d'un compartiment possède une clé et une seule. La combinaison d'un compartiment, d'une clé d'objet et éventuellement d'un ID de version (si la gestion des versions S3 est activée pour le compartiment) identifie de manière unique chaque objet. Vous pouvez donc considérer Amazon S3 comme un mappage de données entre « compartiment + clé + version » et l'objet lui-même.

Chaque objet Amazon S3 peut être adressé de manière unique via la combinaison du point de terminaison du service web, du nom du compartiment, de la clé et, le cas échéant, d'une version. Par exemple, dans l'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`, « `DOC-EXAMPLE-BUCKET` » est le nom du compartiment et « `photos/puppy.jpg` » est la clé.

Pour en savoir plus sur les clés d'objet, consultez Création de noms de clés d'objet.

Gestion des versions S3

Vous pouvez utiliser la gestion des versions S3 pour conserver plusieurs variantes d'un objet dans le même compartiment. Vous pouvez utiliser la gestion des versions S3 pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans vos compartiments. Vous pouvez facilement récupérer les données en cas d'actions involontaires des utilisateurs ou de défaillances des applications.

Pour plus d’informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

ID de version

Lorsque vous activez la gestion des versions S3 pour un compartiment, Amazon S3 génère un ID de version unique pour chaque objet ajouté au compartiment. Les objets qui existaient déjà dans le compartiment au moment où vous activez la gestion des versions ont un ID de version égal à `null`. Si vous modifiez ces objets (ou tout autre) par d'autres opérations, telles que CopyObjectet PutObject, les nouveaux objets obtiennent un ID de version unique.

Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Politique de compartiment

Une politique de compartiment est une politique basée sur les ressources AWS Identity and Access Management (IAM) que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko.

Les stratégies de compartiment utilisent le langage d'access policy basé sur JSON standard dans AWS. Vous pouvez utiliser des stratégies de compartiment pour ajouter ou refuser des autorisations pour les objets d'un compartiment. Les stratégies de compartiment autorisent ou rejettent les demandes basées sur les éléments de la stratégie, y compris le demandeur, les actions S3, les ressources et les aspects ou conditions de la demande (par exemple, l'adresse IP utilisée pour effectuer la demande). Par exemple, vous pouvez créer une stratégie de compartiment qui accorde des autorisations entre comptes pour charger des objets vers un compartiment S3 tout en veillant à ce que le propriétaire du compartiment ait le contrôle total des objets téléchargés. Pour plus d'informations, consultez Exemples de politiques relatives aux compartiments Amazon S3.

Dans votre stratégie de compartiment, vous pouvez utiliser des caractères génériques sur des Amazon Resource Names (ARN) et d'autres valeurs pour accorder des autorisations à un sous-ensemble d'objets. Par exemple, vous pouvez contrôler l'accès aux groupes d'objets qui commencent par un préfixecourant ou se terminent par une extension donnée, comme `.html`.

Points d'accès S3

Les points d'accès Amazon S3 sont nommés points de terminaison réseau avec des stratégies d'accès dédiées qui décrivent comment accéder aux données à l'aide de ce point de terminaison. Les points d'accès sont attachés à des compartiments que vous pouvez utiliser pour effectuer des

opérations sur des objets S3, telles que GetObject etPutObject. Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3.

Chaque point d'accès dispose de sa propre stratégie d'accès. Vous pouvez configurer des paramètres de blocage de l'accès public pour chaque point d'accès. Vous pouvez configurer n'importe quel point d'accès pour accepter uniquement les demandes provenant d'un cloud privé virtuel (VPC) afin de restreindre l'accès aux données Amazon S3 à un réseau privé.

Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Listes de contrôle d'accès (ACL)

Vous pouvez utiliser des ACL pour accorder des autorisations de lecture et d'écriture aux utilisateurs autorisés pour des compartiments et des objets individuels. Chaque compartiment et objet possède une liste ACL qui lui est attachée comme sous-ressource. L'ACL définit le Comptes AWS ou les groupes auxquels l'accès est accordé et le type d'accès. Les ACL représentent un mécanisme de contrôle d'accès qui précède les stratégies IAM. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Régions

Vous pouvez choisir la zone géographique Région AWS dans laquelle Amazon S3 stocke les buckets que vous créez. Vous pouvez choisir une Région pour optimiser la latence, minimiser les coûts ou

répondre aux exigences réglementaires. Les objets stockés dans une région Région AWS ne quittent jamais la région, sauf si vous les transférez ou les répliquez explicitement dans une autre région. Par exemple, les objets stockés dans la Région UE (Irlande) ne la quittent jamais.

> **Note**
>
> Vous pouvez accéder à Amazon S3 et à ses fonctionnalités uniquement dans Régions AWS les zones activées pour votre compte. Pour plus d'informations sur la façon de permettre à une région de créer et de gérer AWS des ressources, consultez Gérer Régions AWS dans le Références générales AWS.

Pour obtenir la liste des points de terminaison et des régions Amazon S3 disponibles, consultez Régions et points de terminaison dans la Références générales AWS.

Modèle de cohérence des données Amazon S3

Amazon S3 assure une forte read-after-write cohérence pour les requêtes PUT et DELETE relatives à l'ensemble des objets de votre compartiment Amazon S3 Régions AWS. Cela s'applique à la fois aux écritures sur de nouveaux objets ainsi qu'aux demandes PUT qui écrasent les objets existants et aux demandes DELETE. En outre, les opérations de lecture sur Amazon S3 Select, les listes de contrôle d'accès (ACL) Amazon S3, les balises d'objet Amazon S3 et les métadonnées d'objet (par exemple l'objet HEAD) sont fortement cohérentes.

Les mises à jour d'une seule clé sont atomiques. Par exemple, si vous faites une demande PUT sur une clé existante d'un thread et exécutez simultanément une demande GET sur la même clé à partir d'un second thread, vous obtiendrez les anciennes données ou les nouvelles données, mais jamais des données partielles ni corrompues.

Amazon S3 garantit une haute disponibilité en répliquant les données sur plusieurs serveurs dans les centres de données AWS . Si une demande PUT aboutit, vos données sont stockées en toute sécurité. Toute lecture (demandes GET ou LIST) initiée après la réception d'une réponse PUT réussie retournera les données écrites par la demande PUT. Voici des exemples de ce comportement :

- Un processus écrit un nouvel objet sur Amazon S3 et dresse immédiatement une liste des clés dans ce compartiment. Le nouvel objet s'affiche dans la liste.

- Un processus remplace un objet existant et tente immédiatement de le lire. Amazon S3 retourne les nouvelles données.
- Un processus efface un objet existant et tente immédiatement de le lire. Amazon S3 ne retourne aucune donnée, car l'objet a été supprimé.
- Un processus efface un objet existant et dresse immédiatement une liste des clés dans ce compartiment. L'objet ne s'affiche pas dans la liste.

> **Note**
>
> - Amazon S3 ne prend pas en charge le verrouillage d'objet pour les auteurs simultanés. Si deux demandes PUT sont effectuées simultanément sur la même clé, la demande indiquant l'horodatage le plus récent est retenue. Si cela pose un problème, vous devrez créer un mécanisme de verrouillage d'objet à votre application.
> - Les mises à jour sont basées sur les clés. Il n'est pas possible d'effectuer des mises à jour atomiques sur plusieurs clés. Par exemple, vous ne pouvez pas mettre à jour une clé qui dépend de la mise à jour d'une autre clé, sauf si vous intégrez cette fonctionnalité dans votre application.

Les configurations de compartiment ont un modèle de cohérence éventuelle. Plus précisément, cela signifie que :

- Si vous supprimez un compartiment et répertoriez immédiatement tous les compartiments, il est possible que le compartiment supprimé figure toujours dans la liste.
- Si vous activez la gestion des versions sur un compartiment pour la première fois, la propagation complète de la modification peut prendre un court laps de temps. Nous vous recommandons d'attendre 15 minutes après l'activation de la gestion des versions avant d'exécuter des opérations d'écriture (demandes PUT ou DELETE) sur les objets du compartiment.

Applications simultanées

Cette section fournit des exemples de comportement à attendre de la part d'Amazon S3 lorsque plusieurs clients écrivent sur les mêmes éléments.

Dans cet exemple, W1 (écriture 1) et W2 (écriture 2) s'achèvent avant le début de R1 (lecture 1) et R2 (lecture 2). Comme S3 est fortement cohérent, R1 et R2 retournent tous les deux `color = ruby`.

Dans l'exemple suivant, W2 ne s'achève pas avant le début de R1. Par conséquent, R1 peut renvoyer `color = ruby` ou `color = garnet`. Cependant, puisque W1 et W2 se terminent avant le début de R2, R2 renvoie `color = garnet`.

Dans le dernier exemple, W2 commence avant que W1 ait reçu un accusé de réception. Par conséquent, ces écritures sont considérées comme simultanées. Amazon S3 utilise la last-writer-wins sémantique en interne pour déterminer quelle écriture est prioritaire. Toutefois, l'ordre dans lequel Amazon S3 reçoit les demandes et l'ordre dans lequel les applications reçoivent des accusés de réception ne peuvent pas être prédits en raison de différents facteurs, comme la latence réseau. Par exemple, W2 peut être initié par une instance Amazon EC2 dans la même Région alors que W1 peut être initié par un hôte plus éloigné. La meilleure façon de déterminer la valeur finale consiste à effectuer la lecture une fois les deux écritures reconnues.

Services connexes

Après avoir chargé vos données dans Amazon S3, vous pouvez les utiliser avec d'autres AWS services. Les services suivants sont ceux que vous êtes susceptibles d'utiliser le plus fréquemment :

- Amazon Elastic Compute Cloud (Amazon EC2) — offre une capacité de calcul évolutive dans le AWS Cloud. L'utilisation d'Amazon EC2 vous dispense d'investir à l'avance dans du matériel et, par conséquent, vous pouvez développer et déployer les applications plus rapidement. Vous pouvez utiliser Amazon EC2 pour lancer autant de serveurs virtuels que nécessaire, configurer la sécurité et les réseaux, et gérer le stockage.
- Amazon EMR : permet aux entreprises, aux chercheurs, aux analystes de données et aux développeurs de traiter de grandes quantités de données de manière simple et économique. Amazon EMR utilise un framework Hadoop hébergé qui s'exécute sur l'infrastructure à l'échelle du web d'Amazon EC2 et d'Amazon S3.
- AWS Snow Family : aide les clients qui doivent mener leurs activités dans des environnements austères, autres que des centres de données, et dans des endroits où la connectivité réseau fait défaut. Vous pouvez utiliser les appareils AWS Snow Family pour accéder localement et à moindre coût au stockage et à la AWS Cloud puissance de calcul des appareils là où une connexion Internet n'est peut-être pas une option.
- AWS Transfer Family : fournit une prise en charge entièrement gérée des transferts de fichiers directement vers et depuis Amazon S3 ou Amazon Elastic File System (Amazon EFS) à l'aide du protocole File Transfer Protocol (SFTP) Secure Shell (SSH), du protocole File Transfer Protocol over SSL (FTPS) et du protocole File Transfer Protocol (FTP).

Accès à Amazon S3

Vous pouvez utiliser Amazon S3 de l'une des façons suivantes :

AWS Management Console

La console est une interface utilisateur Web permettant de gérer Amazon S3 et ses AWS ressources. Si vous vous êtes inscrit à un Compte AWS, vous pouvez accéder à la console Amazon S3 en vous connectant au AWS Management Console et en choisissant S3 sur la page d' AWS Management Console accueil.

AWS Command Line Interface

Vous pouvez utiliser les outils de ligne de AWS commande pour émettre des commandes ou créer des scripts sur la ligne de commande de votre système afin d'effectuer des tâches AWS (y compris S3).

Le AWS Command Line Interface (AWS CLI) fournit des commandes pour un large éventail de Services AWS. AWS CLI Il est pris en charge sur Windows, macOS et Linux. Consultez le AWS Command Line Interface Guide de l'utilisateur pour démarrer. Pour plus d'informations sur les commandes d'Amazon S3, consultez les sections s3api et s3control dans les Références des commandes AWS CLI .

AWS SDK

AWS fournit des SDK (kits de développement logiciel) composés de bibliothèques et d'exemples de code pour différents langages de programmation et plateformes (Java, Python, Ruby, .NET, iOS, Android, etc.). Les AWS SDK constituent un moyen pratique de créer un accès programmatique à S3 et. AWS Amazon S3 est un service REST. Vous pouvez envoyer des demandes à Amazon S3 à l'aide des bibliothèques du AWS SDK, qui encapsulent l'API REST Amazon S3 sous-jacente et simplifient vos tâches de programmation. Par exemple, ils automatisent les tâches comme le calcul de signatures, la signature cryptographique des demandes, la gestion des erreurs et les nouvelles tentatives automatiques de demande. Pour plus d'informations sur les AWS SDK, notamment sur la façon de les télécharger et de les installer, consultez la section Outils pour AWS.

Chaque interaction avec Amazon S3 est authentifiée ou anonyme. Si vous utilisez les AWS SDK, les bibliothèques calculent la signature pour l'authentification à partir des clés que vous fournissez. Pour plus d'informations sur la procédure d'envoi de demandes à Amazon S3, consultez Demandes.

API REST Amazon S3

L'architecture d'Amazon S3 a été conçue de manière à être neutre en termes de langage de programmation et utilise nos interfaces supportées par AWS pour stocker et récupérer des objets. Vous pouvez accéder à S3 et à AWS par programmation à l'aide de l'API REST Amazon S3. L'API REST est une interface HTTP pour Amazon S3. Lorsque vous utilisez l'API REST, vous utilisez des demandes HTTP standard pour créer, récupérer et supprimer des compartiments et des objets.

Pour utiliser l'API REST, vous pouvez choisir n'importe quelle boîte à outils prenant en charge HTTP. Vous pouvez même utiliser un explorateur pour récupérer des objets, si ceux-ci peuvent être lus de manière anonyme.

L'API REST utilise les codes de statut et en-têtes HTTP standard afin de permettre aux explorateurs et boîtes à outils classiques de fonctionner. Dans certains zones, nous avons ajouté des fonctionnalités à HTTP (par exemple, nous avons ajouté des en-têtes afin de permettre le contrôle d'accès). Dans ces cas précis, nous avons fait notre possible pour intégrer cette nouvelle fonctionnalité de sorte qu'elle corresponde à la manière dont HTTP est généralement utilisé.

Si vous faites des appels directs d'API REST dans votre application, vous devez écrire le code pour calculer la signature et l'ajouter à la demande. Pour plus d'informations sur la procédure d'envoi de demandes à Amazon S3, consultez Demandes.

> **Note**
>
> Le support de l'API SOAP via HTTP est obsolète, mais continue d'être disponible sur HTTPS. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Paiement pour Amazon S3

La tarification d'Amazon S3 est conçue de sorte que vous n'avez pas à planifier les besoins en stockage de votre application. La plupart des fournisseurs de stockage exigent que vous achetiez une quantité prédéterminée de capacité de stockage et de transfert réseau. Dans ce scénario, si vous dépassez cette capacité, votre service est fermé ou vous devez payer des coûts supplémentaires élevés. Si vous ne dépassez pas cette limite, vous payerez malgré tout comme si vous l'aviez pleinement utilisée.

Amazon S3 ne facture que ce que vous avez utilisé et il n'y a aucuns frais cachés ou supplémentaires. Ce modèle vous fournit un service à coût variable qui peut évoluer avec votre activité tout en vous offrant les avantages financiers de l'infrastructure. AWS Pour plus d'informations, consultez Tarification Amazon S3.

Lorsque vous vous inscrivez AWS, vous êtes automatiquement Compte AWS inscrit à tous les services AWS, y compris Amazon S3. Toutefois, seuls les services que vous utilisez vous sont facturés. Si vous êtes un nouveau client Amazon S3, vous pouvez commencer à utiliser Amazon S3 gratuitement. Pour plus d'informations, consultez la page sur l'offre gratuite AWS.

Pour consulter votre facture, dirigez-vous vers le Tableau de bord de gestion des coûts et de la facturation dans la AWS Billing and Cost Management console. Pour en savoir plus sur Compte AWS la facturation, consultez le guide de AWS Billing l'utilisateur. Si vous avez des questions concernant AWS la facturation Comptes AWS, contactez le AWS Support.

Conformité PCI DSS

Amazon S3 prend en charge le traitement, le stockage et la transmission des données de cartes bancaires par un commerçant ou un fournisseur de services et a été validé comme étant conforme à la norme PCI (Payment Card Industry) DSS (Data Security Standard). Pour plus d'informations sur la norme PCI DSS, notamment sur la manière de demander une copie du Package de AWS conformité PCI, consultez la section PCI DSS niveau 1.

Premiers pas avec Amazon S3

Vous pouvez démarrer avec Amazon S3 en travaillant avec des compartiments et des objets. Un compartiment est un conteneur d'objets. Un objet est un fichier et toutes les métadonnées qui le décrivent.

Pour stocker un objet dans Amazon S3, vous créez un compartiment, puis téléchargez l'objet dans le compartiment. Lorsque l'objet se trouve dans le compartiment, vous pouvez l'ouvrir, le télécharger et le déplacer. Lorsque vous n'avez plus besoin d'un objet ou d'un compartiment, vous pouvez nettoyer vos ressources.

Avec Amazon S3, vous ne payez que les services que vous utilisez. Pour plus d'informations sur les fonctionnalités et les tarifs d'Amazon S3, consultez Amazon S3. Si vous êtes un nouveau client Amazon S3, vous pouvez commencer à utiliser Amazon S3 gratuitement. Pour plus d'informations, consultez la page sur l'offer gratuite AWS.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Vidéo : Premiers pas avec Amazon S3

Prérequis

Avant de commencer, confirmez que vous avez terminé les étapes de Prérequis : Configuration d'Amazon S3.

Rubriques

- Étapes suivantes

Prérequis : Configuration d'Amazon S3

Lorsque vous vous inscrivez AWS, vous êtes automatiquement Compte AWS inscrit à tous les services AWS, y compris Amazon S3. Seuls les services que vous utilisez vous sont facturés.

Avec Amazon S3, vous ne payez que les services que vous utilisez. Pour plus d'informations sur les fonctionnalités et les tarifs d'Amazon S3, consultez Amazon S3. Si vous êtes un nouveau client Amazon S3, vous pouvez commencer à utiliser Amazon S3 gratuitement. Pour plus d'informations, consultez la page sur l'offer gratuite AWS.

Pour configurer Amazon S3, suivez les étapes décrites dans les sections suivantes.

Lorsque vous vous inscrivez à Amazon S3 AWS et que vous le configurez, vous pouvez éventuellement modifier la langue d'affichage dans le AWS Management Console. Pour de plus amples informations, veuillez consulter la section Modification de la langue de la AWS Management Console du Guide de démarrage de la AWS Management Console .

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et

utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à https://aws.amazon.com/ et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connectez-vous en tant qu'utilisateur disposant d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URL de connexion qui a été envoyée à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAM Identity Center, consultez la section Connexion au portail AWS d'accès dans le guide de l'Connexion à AWS utilisateur.

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

   Pour obtenir des instructions, voir Création d'un ensemble d'autorisations dans le guide de AWS IAM Identity Center l'utilisateur.
2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

   Pour obtenir des instructions, voir Ajouter des groupes dans le guide de AWS IAM Identity Center l'utilisateur.

Étape 1 : Créer votre premier compartiment S3

Une fois inscrit AWS, vous êtes prêt à créer un compartiment dans Amazon S3 à l'aide du AWS Management Console. Dans Amazon S3, chaque objet est stocké dans un compartiment. Avant de pouvoir stocker des données dans Amazon S3, vous devez créer un compartiment.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

> **Note**
>
> Vous n'êtes pas facturé pour la création d'un compartiment. Seuls le stockage d'objets dans le compartiment et le transfert des objets dans et hors du compartiment vous sont facturés. Les frais que vous encourez en appliquant les exemples suivants de ce manuel sont minimes (moins de 1 USD). Pour plus d'informations sur les coûts de stockage, consultez Tarification Amazon S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer un bucket.

   > **Note**
   >
   > Pour limiter la latence et les coûts, et répondre aux exigences légales, choisissez une région proche de vous. Les objets stockés dans une Région ne la quittent jamais, sauf si vous les transférez explicitement vers une autre Région. Pour obtenir la liste d'Amazon S3 Régions AWS, consultez la section sur les Service AWS points de terminaison dans le Référence générale d'Amazon Web Services.

3. Dans le panneau de navigation de gauche, choisissez Compartiments.
4. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
5. Sous Configuration générale, consultez l' Région AWS endroit où votre bucket sera créé.
6. Sous Type de compartiment, sélectionnez Usage général.
7. Pour Nom du compartiment, saisissez le nom de votre compartiment.

   Le nom du compartiment doit présenter les caractéristiques suivantes :

   - Être unique dans une partition. Une partition est un regroupement de Régions. AWS dispose actuellement de trois partitions : `aws` (Régions Standard), `aws-cn` (Régions Chine) et `aws-us-gov` (AWS GovCloud (US) Regions).
   - Il doit comporter entre 3 et 63 caractères.

- Être uniquement composé de lettres minuscules, de chiffres, de points (.) et de traits d'union (-). Pour une meilleure compatibilité, nous vous recommandons d'éviter d'utiliser des points (.) dans les noms de compartiment, à l'exception des compartiments utilisés uniquement pour l'hébergement de sites web statiques.
- Commencer et se terminer par une lettre ou un chiffre.

Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour plus d'informations sur l'attribution de noms à des compartiments, consultez Règles de dénomination de compartiment.

> **Important**
>
> Évitez d'inclure des informations sensibles, notamment des numéros de compte, dans le nom du compartiment. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

8. AWS Management Console vous permet de copier les paramètres d'un bucket existant dans votre nouveau bucket. Si vous ne souhaitez pas copier les paramètres d'un bucket existant, passez à l'étape suivante.

> **Note**
>
> Cette option :
>
> - N'est pas disponible dans le AWS CLI et n'est disponible que dans la console
> - Non disponible pour les compartiments de répertoire
> - Ne copie pas la politique du bucket du bucket existant vers le nouveau bucket

Pour copier les paramètres d'un compartiment existant, sous Copier les paramètres d'un compartiment existant, sélectionnez Choisir un compartiment. La fenêtre Choose bucket s'ouvre. Recherchez le compartiment contenant les paramètres que vous souhaitez copier, puis sélectionnez Choisir un compartiment. La fenêtre Choisir un compartiment se ferme et la fenêtre Créer un compartiment s'ouvre à nouveau.

Sous Copier les paramètres d'un bucket existant, vous pouvez maintenant voir le nom du bucket que vous avez sélectionné. Vous verrez également une option Restaurer les paramètres par

défaut que vous pouvez utiliser pour supprimer les paramètres du bucket copiés. Passez en revue les autres paramètres du compartiment sur la page Créer un compartiment. Vous verrez qu'ils correspondent désormais aux paramètres du bucket que vous avez sélectionné. Vous pouvez passer à la dernière étape.

9. Sous Object Ownership (Propriété de l'objet), pour désactiver ou activer les listes ACL et contrôler la propriété des objets téléchargés dans votre compartiment, sélectionnez l'un des paramètres suivants :

   Listes ACL désactivées

   - Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations d'accès aux données du compartiment S3. Le compartiment utilise des stratégies exclusivement pour définir le contrôle des accès.

     Par défaut, les listes ACL sont désactivées. La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

   Listes ACL activées

   - Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.

     Si vous appliquez le paramètre Propriétaire du compartiment préféré, pour exiger que tous les chargements Amazon S3 incluent la liste ACL prédéfinie `bucket-owner-full-control`, vous pouvez ajouter une politique de compartiment qui autorise uniquement les chargements d'objets utilisant cette liste ACL.

   - Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

> **Note**
>
> Le paramètre par défaut est Propriétaire du compartiment appliqué. Pour appliquer le paramètre par défaut et maintenir les listes ACL désactivées, seule l'autorisation `s3:CreateBucket` est requise. Pour activer les listes ACL, vous devez disposez de l'autorisation `s3:PutBucketOwnershipControls`.

10. Dans Paramètres de blocage de l'accès public pour ce compartiment, choisissez les paramètres Bloquer l'accès public que vous souhaitez appliquer au compartiment.

    Par défaut, les quatre paramètres de blocage de l'accès public sont activés. Nous vous recommandons de maintenir tous les paramètres activés, sauf si vous savez que vous devez en désactiver un ou plusieurs pour votre cas d'utilisation spécifique. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

    > **Note**
    >
    > Pour activer tous les paramètres de blocage de l'accès public, seule l'autorisation `s3:CreateBucket` est requise. Pour désactiver les paramètres de blocage de l'accès public, vous devez disposer de l'autorisation `s3:PutBucketPublicAccessBlock`.

11. (Facultatif) Sous Bucket Versioning (Gestion des versions du compartiment), vous pouvez choisir de conserver les variantes des objets dans votre compartiment. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3.

    Pour désactiver ou activer la gestion des versions sur votre compartiment, choisissez Disable (Désactiver) ou Enable (Activer).

12. (Facultatif) Sous Tags (Balises), vous pouvez choisir d'ajouter des balises à votre compartiment. Les balises sont des paires clé-valeur utilisées pour catégoriser le stockage.

    Pour ajouter une balise de compartiment, saisissez une Key (Clé) et éventuellement une Value (Valeur), puis choisissez Add Tag (Ajouter une balise).

13. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).

14. Pour configurer le chiffrement par défaut, dans Type de chiffrement, choisissez l'une des options suivantes :

- Clés gérées par Amazon S3 (SSE-S3)
- AWS Key Management Service clé (SSE-KMS)

> **Important**
>
> Si vous utilisez l'option SSE-KMS pour votre configuration de chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les AWS KMS quotas et sur la manière de demander une augmentation de quota, consultez la section Quotas dans le guide du AWS Key Management Service développeur.

Les compartiments et les nouveaux objets sont chiffrés à l'aide d'un chiffrement côté serveur avec une clé gérée par Amazon S3 comme niveau de base de configuration du chiffrement. Pour plus d'informations sur le chiffrement par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Pour en savoir plus sur l'utilisation du chiffrement côté serveur Amazon S3 pour chiffrer vos données, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

15. Si vous avez sélectionné CléAWS Key Management Service (SSE-KMS), procédez comme suit :

    a. Sous CléAWS KMS , spécifiez votre clé KMS de l'une des manières suivantes :

    - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

      La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .

    - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
    - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous ne pouvez utiliser que les clés KMS disponibles dans le même compartiment Région AWS que le bucket. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé KMS, puis saisir l'ARN de la clé KMS. Pour plus d'informations sur les autorisations entre comptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service . Pour en savoir plus sur SSE-KMS, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS).
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 prend uniquement en charge les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur l'utilisation AWS KMS avec Amazon S3, consultezUtilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

b. Lorsque vous configurez votre compartiment pour utiliser le chiffrement par défaut avec SSE-KMS, vous pouvez également activer les clés de compartiment S3. Les clés de compartiment S3 réduisent le coût du chiffrement en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

   Pour utiliser les clés de compartiment S3, sous la Clé de compartiment, choisissez Activer.

16. (Facultatif) Si vous souhaitez activer le verrouillage des objets S3, procédez comme suit :

    a. Choisissez Advanced Settings (Paramètres avancés).

    > **Important**
    >
    > L'activation du verrouillage d'objet active également la gestion des versions pour le compartiment. Après l'avoir activé, vous devez configurer les paramètres de conservation et de mise en suspens juridique par défaut du verrouillage d'objets pour protéger les nouveaux objets contre la suppression ou l'écrasement.

    b. Pour activer le verrouillage d'objets, choisissez Enable (Activer), lisez l'avertissement qui s'affiche et confirmez-le.

    Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

    > **Note**
    >
    > Pour créer un compartiment prenant en charge le verrouillage d'objets, vous devez disposer des autorisations suivantes : `s3:CreateBucket`, `s3:PutBucketVersioning` et `s3:PutBucketObjectLockConfiguration`.

17. Choisissez Créer un compartiment.

Vous avez créé un compartiment dans Amazon S3.

Étape suivante

Pour ajouter un objet à votre compartiment, veuillez consulter Étape 2 : Charger un objet dans votre compartiment.

Étape 2 : Charger un objet dans votre compartiment

Après avoir créé un compartiment dans Amazon S3, vous êtes prêt à charger un objet dans celui-ci. Un objet peut désigner n'importe quel type de fichier : un fichier texte, une photo, une vidéo, etc.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Pour charger un objet dans un compartiment

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiment), choisissez le nom du compartiment dans lequel vous souhaitez charger votre objet.
3. Sous l'onglet Objets de votre compartiment, choisissez Charger.
4. Sous Fichiers et dossiers, choisissez Ajouter des fichiers.
5. Choisissez un fichier à charger, puis choisissez Ouvrir.
6. Sélectionnez Charger.

Vous avez réussi à charger un objet dans votre compartiment.

Étape suivante

Pour afficher votre objet, veuillez consulter Étape 3 :Télécharger un objet.

Étape 3 :Télécharger un objet

Lorsque vous chargez un objet dans un compartiment, vous pouvez afficher des informations sur votre objet et télécharger l'objet sur votre ordinateur local.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Utilisation de la console S3

Cette section explique comment utiliser la console Amazon S3 pour télécharger un objet depuis un compartiment S3.

> **Note**
>
> - Vous ne pouvez télécharger qu'un seul objet à la fois.
> - Si vous utilisez la console Amazon S3 pour télécharger un objet dont le nom de clé se termine par un point (`.`), celui-ci est supprimé du nom de clé de l'objet téléchargé. Pour conserver le point à la fin du nom de l'objet téléchargé, vous devez utiliser l'AWS Command Line Interface (AWS CLI), les kits AWS SDK ou l'API REST Amazon S3.

Pour télécharger un objet à partir d'un compartiment S3

1. Connectez-vous à AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment à partir duquel vous souhaitez télécharger un objet.
3. Vous pouvez télécharger un objet à partir d'un compartiment S3 de l'une des façons suivantes :
   - Cochez la case à côté de l'objet et choisissez Télécharger. Si vous souhaitez télécharger l'objet dans un dossier spécifique, dans le menu Actions, choisissez Télécharger en tant que.
   - Si vous souhaitez télécharger une version spécifique de l'objet, activez Afficher les versions (en regard de la zone de recherche). Cochez la case à côté de la version de l'objet de votre choix, puis choisissez Télécharger. Si vous souhaitez télécharger l'objet dans un dossier spécifique, dans le menu Actions, choisissez Télécharger en tant que.

Vous avez téléchargé votre objet avec succès.

Étape suivante

Pour copier et coller votre objet dans Amazon S3, veuillez consulter Étape 4 : Copiez votre objet dans un dossier.

Étape 4 : Copiez votre objet dans un dossier

Vous avez déjà ajouté un objet dans un compartiment et téléchargé l'objet. Maintenant, vous créez un dossier et copiez l'objet et collez-le dans le dossier.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Pour copier un objet dans un dossier

1. Dans la liste Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Create folder (Créer un dossier) et configurez le nouveau dossier :
   a. Entrez un nom de dossier (par exemple, `favorite-pics`).
   b. Pour le paramètre de chiffrement du dossier, choisissez Désactiver.
   c. Choisissez Enregistrer.
3. Accédez au compartiment ou au dossier Amazon S3 contenant les objets que vous souhaitez copier.
4. Cochez la case située à gauche des noms des objets que vous souhaitez copier.
5. Choisissez Actions et Copy (Copier) dans la liste des options qui s'affiche.

   Vous pouvez également choisir Copy (Copier) parmi les options en haut à droite.
6. Choisissez le dossier de destination.
   a. Choisissez Parcourir S3.
   b. Cliquez sur le bouton d'option situé à gauche du nom du dossier.

      Pour naviguer dans un dossier et choisir un sous-dossier comme destination, choisissez le nom du dossier.
   c. Choisissez Choose destination (Choisir une destination).

Le chemin d'accès au dossier de destination apparaît dans la zone Destination. Dans Destination, vous pouvez alternativement entrer votre chemin de destination, par exemple, s3://*nom-compartiment*/*nom-dossier*/.

7. En bas à droite, choisissez Copy (Copier).

   Amazon S3 copie vos objets dans le dossier de destination.

Étape suivante

Pour supprimer un objet et un compartiment dans Amazon S3, veuillez consulter Étape 5 : Supprimer vos objets et votre compartiment.

Étape 5 : Supprimer vos objets et votre compartiment

Lorsque vous n'avez plus besoin d'un objet ou d'un compartiment, nous vous recommandons de les supprimer pour éviter des frais supplémentaires. Si vous avez terminé cette démonstration de démarrage en tant qu'exercice de formation et que vous ne prévoyez pas d'utiliser votre compartiment ou vos objets, nous recommandons de supprimer votre compartiment et vos objets afin que les frais ne s'accumulent plus.

Avant de supprimer votre compartiment, vous devez vider celui-ci ou supprimer les objets qui se trouvent dans le compartiment. Une fois que vous avez supprimé vos objets et votre compartiment, ils ne sont plus disponibles.

Si vous souhaitez continuer à utiliser le même nom de compartiment, nous vous recommandons de supprimer les objets ou de vider le compartiment, mais sans le supprimer. Une fois que vous avez supprimé un compartiment, le nom devient disponible et peut être réutilisé. Toutefois, un autre Compte AWS peut créer un compartiment portant le même nom avant que vous n'ayez l'occasion de réutiliser celui-ci.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Suppression d'un objet

Si vous souhaitez choisir les objets que vous supprimez sans vider tous les objets de votre compartiment, vous pouvez supprimer un objet.

1. Dans la liste Compartiments, choisissez le nom du compartiment à partir duquel vous souhaitez supprimer un objet.
2. Sélectionnez l'objets que vous voulez supprimer.
3. Choisissez Supprimer dans les options affichées en haut à droite.
4. Sur la page Supprimer des objets, tapez **`delete`** pour confirmer la suppression de vos objets.
5. Choisissez Supprimer les objets.

Vider votre compartiment

Si vous envisagez de supprimer votre compartiment, vous devez d'abord vider celui-ci, ce qui supprimera tous les objets du compartiment.

Pour vider un compartiment :

1. Dans la liste Compartiments, sélectionnez le compartiment à vider, puis choisissez Vider.
2. Pour confirmer que vous souhaitez vider le compartiment et supprimer tous les objets qu'il contient, dans Empty bucket (Vider le compartiment), saisissez **`permanently delete`**.

   > **Important**
   >
   > Cette opération ne peut pas être annulée. Les objets ajoutés au compartiment pendant le vidage de celui-ci seront supprimés.

3. Pour vider le compartiment et supprimer tous les objets qu'il contient, choisissez Vider.

Une page Empty bucket: Status (Vider le compartiment : statut) s'ouvre et vous permet de consulter un résumé des suppressions d'objets qui ont réussi et échoué.

4. Pour revenir à votre liste de compartiments, choisissez Quitter.

Suppression de votre compartiment

Après avoir vidé votre compartiment ou supprimé tous les objets de votre compartiment, vous pouvez supprimer celui-ci.

1. Pour supprimer un compartiment, dans la liste Compartiments, sélectionnez le compartiment.
2. Sélectionnez Delete.
3. Pour confirmer la suppression, dans Delete bucket (Supprimer le compartiment), entrez le nom du compartiment.

> **Important**
>
> La suppression d'un compartiment ne peut pas être annulée. Les noms de compartiment sont uniques. Si vous supprimez votre compartiment, un autre utilisateur AWS peut utiliser le nom qui lui était attribué. Si vous souhaitez continuer à utiliser le même nom de compartiment, ne supprimez pas le compartiment. Au lieu de cela, videz et conservez le compartiment.

4. Pour supprimer votre compartiment, choisissez Supprimer le compartiment.

Étapes suivantes

Dans les exemples précédents, vous avez appris à exécuter certaines tâches Amazon S3 élémentaires.

Les rubriques suivantes expliquent les chemins d'apprentissage que vous pouvez utiliser pour mieux comprendre Amazon S3 afin de pouvoir l'implémenter dans vos applications.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Comprendre les cas d'utilisation courants

Vous pouvez utiliser Amazon S3 pour prendre en charge votre cas d'utilisation particulier. La Bibliothèque de solutions AWS et le blog AWSfournissent des informations et des didacticiels propres au cas d'utilisation. Voici quelques cas d'utilisation courants d'Amazon S3 :

- Sauvegarde et stockage – Utilisez les fonctions de gestion du stockage Amazon S3 pour gérer les coûts, respecter les exigences réglementaires, réduire la latence et enregistrer plusieurs copies distinctes de vos données à des fins de respect des exigences de conformité.
- Hébergement d'applications – Déployez, installez et gérez des applications Web fiables, hautement évolutives et économiques. Par exemple, vous pouvez configurer votre compartiment Amazon S3 pour héberger un site web statique. Pour plus d'informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.
- Hébergement de contenus multimédias – Développez une infrastructure hautement disponible qui héberge des chargements et des téléchargements de vidéos, photos ou musique.
- Livraison de logiciels – Hébergez vos applications logicielles afin que vos clients puissent les télécharger.

Contrôlez l'accès à vos compartiments et à vos objets

Amazon S3 fournit plusieurs fonctionnalités et outils de sécurité. Pour avoir une présentation, consultez Gestion des accès.

Par défaut, les compartiments S3 et les objets sont privés. Vous avez accès uniquement aux ressources S3 que vous créez. Vous pouvez utiliser les fonctions suivantes pour accorder des autorisations de ressources granulaires qui prennent en charge votre cas d'utilisation particulier ou pour auditer les autorisations de vos ressources Amazon S3.

- Bloquer l'accès public S3 – Bloquer l'accès public des compartiments S3 et des objets. Par défaut, les paramètres de blocage de l'accès public sont activés au niveau du compartiment.
- AWS Identity and Access Management Identités (IAM) — Utilisez IAM ou créez AWS IAM Identity Center des identités IAM Compte AWS pour gérer l'accès à vos ressources Amazon S3. Par exemple, vous pouvez utiliser IAM avec Amazon S3 pour contrôler le type d'accès d'un utilisateur ou d'un groupe d'utilisateurs à un compartiment Amazon S3 qui vous Compte AWS appartient. Pour obtenir plus d'informations sur les identités IAM et sur les bonnes pratiques, consultez la section IAM Identities (users, user groups, and roles) [Identités IAM (utilisateurs, groupes d'utilisateurs et rôles)] dans le Guide de l'utilisateur IAM.
- Politiques de compartiment – Utilisez un langage de politique basé sur IAM pour configurer les autorisations basées sur les ressources de vos compartiments S3 et des objets qu'ils contiennent.
- Listes de contrôle d'accès (ACL) – Accordez des autorisations de lecture et d'écriture pour des compartiments et des objets individuels aux utilisateurs autorisés. En règle générale, nous recommandons d'utiliser des politiques basées sur les ressources S3 (politiques de compartiment et politiques de point d'accès) ou des politiques d'utilisateur IAM pour le contrôle d'accès à la place des listes ACL. Les politiques constituent une option de contrôle d'accès simplifiée et plus flexible. Les politiques de compartiment et de point d'accès vous permettent de définir des règles s'appliquant de manière générale à toutes les demandes adressées à vos ressources Amazon S3. Pour plus d'informations sur les cas spécifiques où vous devriez utiliser des listes ACL plutôt que des politiques basées sur les ressources ou des politiques d'utilisateur IAM, consultez Identity and Access Management pour Amazon S3.
- Propriété d'objets S3 : prenez possession de chaque objet présent dans votre compartiment, afin de simplifier la gestion des accès aux données stockées dans Amazon S3. La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour désactiver ou activer les listes ACL. Par défaut, les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets présents dans

le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès.

- Analyseur d'accès IAM pour S3 : évaluez et contrôlez vos politiques d'accès aux compartiments S3, en veillant à ce que ces politiques fournissent uniquement l'accès prévu à vos ressources S3.

Gérer et surveiller votre stockage

- Gérer votre stockage – Après avoir créé des compartiments et chargé des objets dans Amazon S3, vous pourrez gérer votre stockage d'objets. Par exemple, vous pourrez utiliser la gestion des versions S3 et la réplication S3 pour la reprise après sinistre, le cycle de vie S3 pour gérer les coûts de stockage et le verrouillage d'objets S3 pour répondre aux exigences de conformité.
- Surveillance de votre stockage : la surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon S3 et de vos AWS solutions. Vous pourrez surveiller l'activité et les coûts de stockage. En outre, nous vous recommandons de recueillir les données de surveillance de toutes les parties de votre solution AWS afin de pouvoir déboguer plus facilement une éventuelle défaillance à plusieurs points.
- Analytique et informations – Vous pouvez utiliser l’analytique et les informations dans Amazon S3 pour comprendre, analyser et optimiser votre utilisation du stockage. Par exemple, utilisez Amazon S3 Storage Lens pour comprendre, analyser et optimiser votre stockage. S3 Storage Lens offre plus de 29 métriques d'utilisation et d'activité ainsi que des tableaux de bord interactifs vous permettant d'agréger des données pour l'ensemble de votre organisation, des comptes spécifiques, des Régions, des compartiments ou des préfixes. Utilisez Analyse de classe de stockage pour analyser les modèles d'accès au stockage afin de décider lorsqu'il est temps de déplacer vos données vers une classe de stockage plus économique.

Développer avec Amazon S3

Amazon S3 est un service REST. Vous pouvez envoyer des demandes à Amazon S3 à l'aide de l'API REST ou des bibliothèques du AWS SDK, qui encapsulent l'API REST Amazon S3 sous-jacente, simplifiant ainsi vos tâches de programmation. Vous pouvez également utiliser l' AWS Command Line Interface (AWS CLI) pour effectuer des appels d'API Amazon S3. Pour plus d’informations, consultez Demandes.

L'API REST Amazon S3 est une interface HTTP vers Amazon S3. Lorsque vous utilisez l'API REST, vous utilisez des demandes HTTP standard pour créer, récupérer et supprimer des compartiments et des objets. Pour utiliser l'API REST, vous pouvez choisir n'importe quelle boîte à outils prenant

en charge HTTP. Vous pouvez même utiliser un explorateur pour récupérer des objets, si ceux-ci peuvent être lus de manière anonyme. Pour plus d'informations, consultez Développer avec Amazon S3 à l'aide de l'API REST.

Pour vous aider à développer des applications utilisant le langage de votre choix, nous fournissons les ressources suivantes.

AWS CLI

Vous pouvez accéder aux fonctions d'Amazon S3 à l'aide de AWS CLI. Pour télécharger et configurer le AWS CLI, voirDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

AWS CLI Il fournit deux niveaux de commandes pour accéder à Amazon S3 : les commandes de haut niveau (s3) et les commandes de niveau API (s3api et s3control). Les commandes S3 de niveau élevé simplifient l'exécution des tâches courantes, telles que la création, la manipulation et la suppression d'objets et de compartiments. Les commandes s3api et s3control exposent l'accès direct à toutes les opérations d'API Amazon S3, que vous pouvez utiliser pour effectuer des opérations avancées qui pourraient ne pas être possibles avec les commandes de niveau élevé.

Pour obtenir la liste des AWS CLI commandes Amazon S3, consultez s3, s3api et s3control.

AWS SDK et explorateurs

Vous pouvez utiliser les AWS SDK lorsque vous développez des applications avec Amazon S3. Les kits SDK AWS simplifient vos tâches de programmation en encapsulant l'API REST sous-jacente. Les SDK AWS mobiles et la bibliothèque JavaScript Amplify sont également disponibles pour créer des applications mobiles et Web connectées à l'aide de. AWS

Outre les AWS SDK, des AWS explorateurs sont disponibles pour Visual Studio et Eclipse pour Java IDE. Dans ce cas, les SDK et les explorateurs sont regroupés sous forme de boîtes à outils. AWS

Pour plus d'informations, consultez Développement avec Amazon S3 à l'aide des AWS SDK.

Exemples de codes et bibliothèques

Le centre pour développeurs AWS et le catalogue d'exemples de codes AWS proposent des exemples de codes et des bibliothèques spécialement écrits pour Amazon S3. Vous pouvez utiliser ces exemples de codes afin de comprendre comment mettre en œuvre l'API Amazon S3. Vous pouvez également consulter la Référence des API Amazon Simple Storage Service pour comprendre les opérations de l'API Amazon S3 en détails.

Apprendre à partir de tutoriels

Vous pouvez commencer par des step-by-step didacticiels pour en savoir plus sur Amazon S3. Ces sont conçus pour un environnement de travaux pratiques et ils utilisent des noms de sociétés fictifs, des noms d'utilisateur fictifs, etc. Leur objectif consiste à fournir des instructions générales. Ils ne sont pas conçus pour être utilisés directement dans votre environnement de production sans être préalablement vérifiés et adaptés soigneusement aux besoins uniques de votre organisation.

Premiers pas

- Tutoriel : stockage et récupération d'un fichier avec Amazon S3
- Tutoriel : démarrer avec Amazon S3 Intelligent-Tiering
- Tutoriel : démarrer avec les classes de stockage Amazon S3 Glacier

Optimisation des coûts de stockage

- Tutoriel : démarrer avec Amazon S3 Intelligent-Tiering
- Tutoriel : démarrer avec les classes de stockage Amazon S3 Glacier
- Tutoriel : optimisation des coûts et amélioration de la visibilité avec S3 Storage Lens

Gestion du stockage

- Tutorial: Getting started with Amazon S3 Multi-Region Access Points (Didacticiel : Débuter à l'aide des points d'accès multi-régions Amazon S3)
- Tutoriel : Réplication d'objets existants dans vos compartiments Amazon S3 avec la réplication par lot S3

Hébergement de vidéos et de sites web

- Tutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53
- Didacticiel : configuration d'un site web statique sur Amazon S3
- Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53

Traitement des données

- Didacticiel : Transformation de données pour votre application avec S3 Object Lambda
- Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend
- Tutoriel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération
- Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Protection des données

- Tutoriel : vérifier l'intégrité des données dans Amazon S3 avec des totaux de contrôle supplémentaires
- Tutoriel : Réplication de données au sein et entre les deux à Régions AWS l'aide de S3 Replication
- Tutoriel : protection des données sur Amazon S3 contre les suppressions accidentelles ou les bogues d'application à l'aide de la gestion des versions S3, du verrouillage d'objets S3 et de la réplication S3
- Tutoriel : Réplication d'objets existants dans vos compartiments Amazon S3 avec la réplication par lot S3

Explorer la formation et le support

Vous pouvez apprendre auprès d' AWS experts pour améliorer vos compétences et obtenir l'assistance d'experts pour atteindre vos objectifs.

- Formation – Les ressources de formation offrent une approche pratique de l'apprentissage d'Amazon S3. Pour en savoir plus, consultez Formation et certification AWS et AWS online tech talks.
- Forums de discussion – Sur le forum, vous pouvez consulter des publications pour comprendre ce que vous pouvez faire ou non avec Amazon S3. Vous pouvez également poser des questions. Pour en savoir plus, consultez Forums de discussion.
- Assistance technique – Si vous avez d'autres questions, vous pouvez contacter l'Assistance technique.

Didacticiels

Les didacticiels suivants présentent end-to-end des procédures complètes pour les tâches courantes d'Amazon S3. Ces sont conçus pour un environnement de travaux pratiques et ils utilisent des noms de sociétés fictifs, des noms d'utilisateur fictifs, etc. Leur objectif consiste à fournir des instructions générales. Ils ne sont pas conçus pour être utilisés directement dans votre environnement de production sans être préalablement vérifiés et adaptés soigneusement aux besoins uniques de votre organisation.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Premiers pas

- Tutoriel : stockage et récupération d'un fichier avec Amazon S3
- Tutoriel : démarrer avec Amazon S3 Intelligent-Tiering
- Tutoriel : démarrer avec les classes de stockage Amazon S3 Glacier

Optimisation des coûts de stockage

- Tutoriel : démarrer avec Amazon S3 Intelligent-Tiering
- Tutoriel : démarrer avec les classes de stockage Amazon S3 Glacier
- Tutoriel : optimisation des coûts et amélioration de la visibilité avec S3 Storage Lens

Gestion du stockage

- Tutorial: Getting started with Amazon S3 Multi-Region Access Points (Didacticiel : Débuter à l'aide des points d'accès multi-régions Amazon S3)
- Tutoriel : Réplication d'objets existants dans vos compartiments Amazon S3 avec la réplication par lot S3

Hébergement de vidéos et de sites web

- Tutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53
- Didacticiel : configuration d'un site web statique sur Amazon S3
- Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53

Traitement des données

- Didacticiel : Transformation de données pour votre application avec S3 Object Lambda
- Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend
- Tutoriel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération
- Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Protection des données

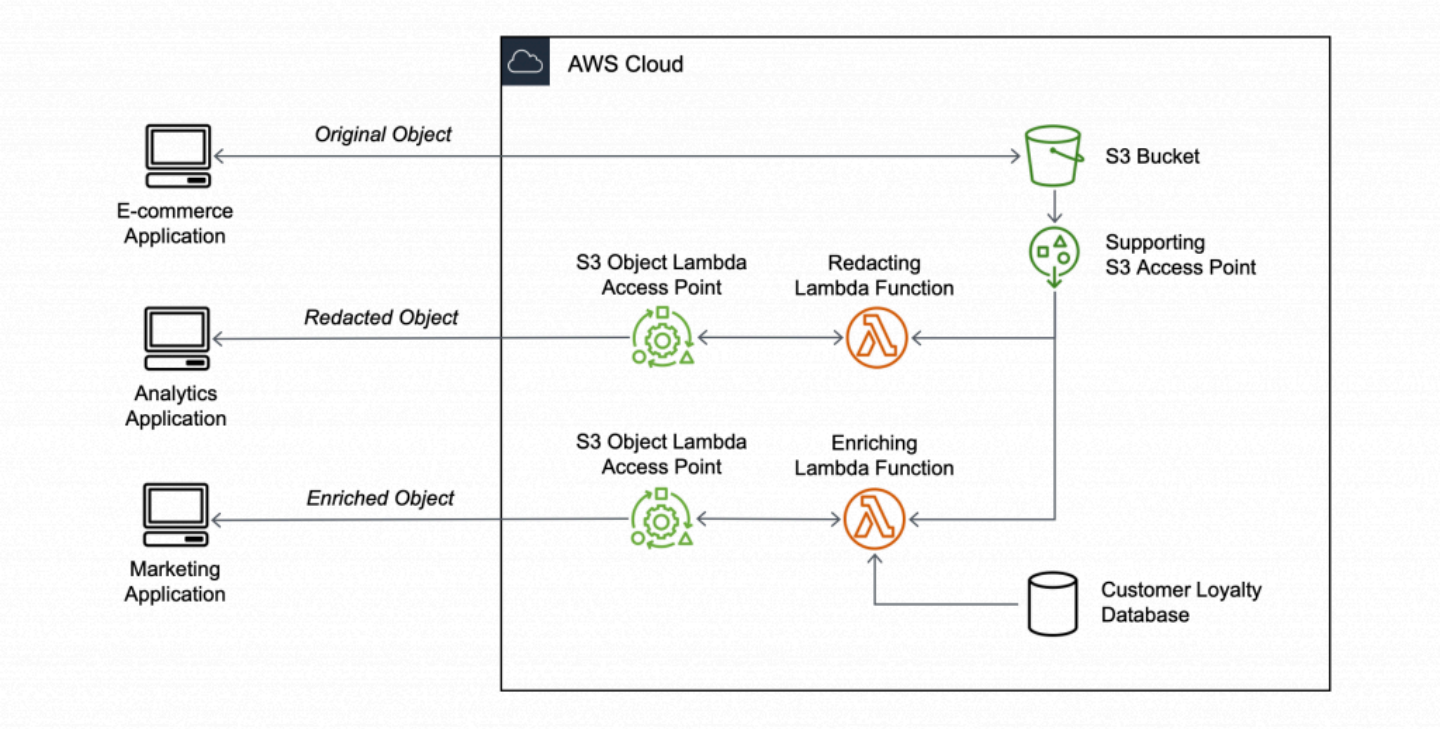
- Tutoriel : vérifier l'intégrité des données dans Amazon S3 avec des totaux de contrôle supplémentaires
- Tutoriel : Réplication de données au sein et entre les deux à Régions AWS l'aide de S3 Replication
- Tutoriel : protection des données sur Amazon S3 contre les suppressions accidentelles ou les bogues d'application à l'aide de la gestion des versions S3, du verrouillage d'objets S3 et de la réplication S3
- Tutoriel : Réplication d'objets existants dans vos compartiments Amazon S3 avec la réplication par lot S3

Didacticiel : Transformation de données pour votre application avec S3 Object Lambda

Lorsque vous stockez des données dans Amazon S3, vous pouvez facilement les partager pour utilisation par plusieurs applications. Cependant, chaque application peut avoir des exigences de

format de données uniques et nécessiter une modification ou un traitement de vos données pour un cas d'utilisation particulier. Par exemple, un jeu de données créé par une application de commerce électronique peut inclure des données à caractère personnel identifiables (DCPI). Lorsque les mêmes données sont traitées à des fins analytiques, ces DCPI ne sont pas nécessaires et doivent être supprimées. Toutefois, si le même jeu de données est utilisé pour une campagne marketing, vous devrez peut-être enrichir les données avec des détails supplémentaires, comme des informations provenant de la base de données de fidélisation des clients.

Avec S3 Object Lambda, vous pouvez ajouter votre propre code pour traiter les données récupérées à partir de S3 avant de les renvoyer vers une application. Plus précisément, vous pouvez configurer une AWS Lambda fonction et l'associer à un point d'accès Lambda S3 Object. Lorsqu'une application envoie des demandes GET S3 standard via le point d'accès S3 Object Lambda, la fonction Lambda indiquée est appelée pour traiter toutes les données extraites d'un compartiment S3 via le point d'accès S3 de prise en charge. Ensuite, le point d'accès S3 Object Lambda renvoie le résultat transformé à l'application. Vous pouvez créer et exécuter vos propres fonctions Lambda personnalisées, en adaptant la transformation des données S3 Object Lambda à votre cas d'utilisation particulier, sans devoir modifier vos applications.



Objectif

Dans ce didacticiel, vous apprendrez à ajouter du code personnalisé aux demandes GET S3 standard pour modifier l'objet demandé récupéré à partir de S3 afin que l'objet réponde aux besoins

du client ou de l'application demandeur. Plus précisément, vous apprendrez à transformer en majuscules tout le texte de l'objet original stocké dans S3 grâce à S3 Object Lambda.

Rubriques

Prérequis

Avant de commencer ce didacticiel, vous devez disposer d'un utilisateur Compte AWS auquel vous pouvez vous connecter en tant qu'utilisateur AWS Identity and Access Management (IAM) avec les autorisations appropriées. Vous devez également installer Python 3.8 ou version ultérieure.

Sous-étapes

Créez un utilisateur IAM avec des autorisations dans votre Compte AWS (console)

Vous pouvez créer un utilisateur IAM pour le didacticiel. Pour terminer ce didacticiel, votre utilisateur IAM doit joindre les politiques IAM suivantes pour accéder aux AWS ressources pertinentes et effectuer des actions spécifiques. Pour en savoir plus sur la création d'un utilisateur IAM, consultez Créer des utilisateurs IAM (console) dans le guide de l'utilisateur IAM.

Votre utilisateur IAM requiert les politiques suivantes :

- AmazonS3 FullAccess — Accorde des autorisations pour toutes les actions Amazon S3, y compris les autorisations pour créer et utiliser un point d'accès Object Lambda.
- AWSLambda_FullAccess— Accorde des autorisations à toutes les actions Lambda.
- IAM FullAccess — Accorde des autorisations à toutes les actions IAM.
- IAM AccessAnalyzerReadOnlyAccess — Accorde l'autorisation de lire toutes les informations d'accès fournies par IAM Access Analyzer.
- CloudWatchLogsFullAccess— Accorde un accès complet aux CloudWatch journaux.

> **Note**
>
> Pour des raisons de simplicité, ce didacticiel crée et utilise un utilisateur IAM. Après avoir terminé ce didacticiel, n'oubliez pas de Supprimer l'utilisateur IAM. Pour une utilisation en production, nous vous recommandons de suivre les Meilleures pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM. Une bonne pratique requiert que les utilisateurs humains utilisent une fédération avec un fournisseur d'identité pour accéder à AWS avec des informations d'identification temporaires. Une autre bonne pratique consiste à exiger que les charges de travail utilisent des informations d'identification temporaires avec des rôles IAM pour accéder à AWS. Pour en savoir plus sur l'utilisation AWS IAM Identity Center pour créer des utilisateurs avec des informations d'identification temporaires, voir Getting started dans le guide de AWS IAM Identity Center l'utilisateur.
> Ce didacticiel utilise des politiques gérées AWS à accès complet. Pour utilisation en production, nous vous recommandons d'accorder uniquement les autorisations minimales nécessaires à votre cas d'utilisation, conformément aux Bonnes pratiques de sécurité.

Installez Python 3.8 ou version ultérieure sur votre machine locale

Utilisez la procédure suivante pour installer Python 3.8 ou version ultérieure sur votre machine locale. Pour obtenir des instructions sur l'installation, consultez la page Télécharger Python dans le Guide du débutant de Python.

1. Ouvrez votre terminal local ou votre coquille locale, puis exécutez la commande suivante afin de déterminer si Python est déjà installé et, si oui, quelle version est installée.

   ```
   python --version
   ```

2. Si Python 3.8 ou version ultérieure n'est pas installé, téléchargez le programme d'installation officiel de Python 3.8 ou version ultérieure qui convient à votre machine locale.
3. Exécutez le programme d'installation en double-cliquant sur le fichier téléchargé, puis suivez les étapes pour achever l'installation.

   Pour les utilisateurs Windows, choisissez Ajouter Python 3.X au PATH dans l'assistant d'installation avant de choisir Installer maintenant.
4. Redémarrez votre terminal en le fermant et en le rouvrant.
5. Exécutez la commande suivante pour vérifier que Python 3.8 ou version ultérieure est installé correctement.

   Pour les utilisateurs macOS, exécutez cette commande :

   ```
   python3 --version
   ```

   Pour les utilisateurs Windows, exécutez cette commande :

   ```
   python --version
   ```
6. Exécutez les commandes suivantes pour vérifier que le gestionnaire de paquets pip3 est installé. Si vous voyez un numéro de version pip et Python 3.8 ou version ultérieure dans la réponse de la commande, cela signifie que le gestionnaire de paquets pip3 est installé correctement.

   ```
   pip --version
   ```

Étape 1 : Créer un compartiment S3

Créez un compartiment pour stocker les données originales que vous prévoyez de transformer.

Pour créer un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.

4. Pour Nom du compartiment, saisissez un nom (par exemple, **`tutorial-bucket`**) pour votre compartiment.

   Pour en savoir plus sur les règles d'attribution de noms des compartiments Amazon S3, consultez Règles de dénomination de compartiment.

5. Pour Région, choisissez l' Région AWS endroit où vous souhaitez que le compartiment réside.

   Pour en savoir plus sur les régions des compartiments, consultez Présentation des compartiments.

6. Pour Paramètres de blocage de l'accès public à ce compartiment, conservez les paramètres par défaut (Bloquer tout accès public est activé).

   Nous vous recommandons de laisser tous les paramètres de blocage de l'accès public activés, sauf si vous devez en désactiver un ou plusieurs pour votre cas d'utilisation. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

7. Pour les paramètres restants, conservez les paramètres par défaut.

   (Facultatif) Si vous souhaitez configurer des paramètres de compartiment supplémentaires pour votre cas d'utilisation particulier, consultez Créer un compartiment.

8. Choisissez Créer un compartiment.

Étape 2 : Charger un fichier dans le compartiment S3

Chargez un fichier texte dans le compartiment S3. Ce fichier texte contient les données originales que vous transformerez en majuscules plus tard dans ce didacticiel.

Par exemple, vous pouvez charger un fichier `tutorial.txt` qui contient le texte suivant :

```
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
returning it to an application.
```

Charger un fichier dans un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.

3. Dans la liste Compartiments, choisissez le nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**) pour y charger votre fichier.
4. Sous l'onglet Objets de votre compartiment, choisissez Charger.
5. Dans la page Charger, sous Fichiers et dossiers, choisissez Ajouter des fichiers.
6. Choisissez un fichier à charger, puis choisissez Ouvrir. Par exemple, vous pouvez charger le fichier `tutorial.txt` mentionné précédemment.
7. Choisissez Charger.

Étape 3 : Créer un point d'accès S3

Pour utiliser un point d'accès S3 Object Lambda afin d'accéder aux données originales et de les transformer, vous devez créer un point d'accès S3 et l'associer au compartiment S3 que vous avez créé à l'étape 1. Le point d'accès doit se trouver dans le même Région AWS emplacement que les objets que vous souhaitez transformer.

Plus loin dans ce didacticiel, vous utiliserez ce point d'accès comme point d'accès de prise en charge pour votre point d'accès Object Lambda.

Créer un point d'accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation, choisissez Points d'accès.
3. Dans la page Points d'accès, choisissez Créer un point d'accès.
4. Dans le champ Nom du point d'accès, saisissez le nom (par exemple, **`tutorial-access-point`**) du point d'accès.

   Pour en savoir plus sur l'attribution de noms aux points d'accès, consultez Règles relatives à l'attribution de noms pour les points d'accès Amazon S3.
5. Dans Nom du compartiment, saisissez le nom du compartiment que vous avez créé à l'étape 1(par exemple, **`tutorial-bucket`**). S3 attache le point d'accès à ce compartiment.

   (Facultatif) Vous pouvez choisir Parcourir S3 pour parcourir et rechercher les compartiments inclus dans votre compte. Si vous choisissez Parcourir S3, choisissez le compartiment souhaité, puis choisissez Choisir un chemin pour renseigner le champ Nom du compartiment avec le nom de ce compartiment.

6. Pour Origine du réseau, choisissez Internet.

   Pour en savoir plus sur les origines de réseau des points d'accès, consultez Création de points d'accès restreints à un virtual private cloud.

7. Tous les paramètres de blocage d'accès public seront activés par défaut à votre point d'accès. Nous vous recommandons de conserver l'option Bloquer tous les accès publics activée.

   Pour en savoir plus, consultez Gestion de l'accès public aux points d'accès.

8. Pour tous les autres paramètres de point d'accès, conservez les paramètres par défaut.

   (Facultatif) Vous pouvez modifier les paramètres du point d'accès afin de prendre en charge votre cas d'utilisation. Pour ce didacticiel, nous vous recommandons de conserver les paramètres par défaut.

   (Facultatif) Si vous devez gérer l'accès à votre point d'accès, vous pouvez indiquer une politique de point d'accès. Pour plus d'informations, consultez Exemples de stratégie de point d'accès.

9. Choisissez Créer un point d'accès.

Étape 4 : Créer une fonction Lambda

Pour transformer des données originales, créez une fonction Lambda à utiliser avec votre point d'accès S3 Object Lambda.

Sous-étapes

- Écrire le code d'une fonction Lambda et créer un package de déploiement avec un environnement virtuel
- Créer une fonction Lambda à l'aide d'un rôle d'exécution (console)
- Déployez le code de votre fonction Lambda avec les archives du fichier .zip et configurez la fonction Lambda (console)

Écrire le code d'une fonction Lambda et créer un package de déploiement avec un environnement virtuel

1. Sur votre ordinateur local, créez un dossier avec le nom du dossier `object-lambda` afin que l'environnement virtuel puisse l'utiliser ultérieurement dans ce didacticiel.

2. Dans le dossier `object-lambda`, créez un fichier avec une fonction Lambda qui change tout le texte de l'objet original en majuscules. Par exemple, vous pouvez utiliser la fonction suivante écrite en langage Python. Enregistrez cette fonction dans un fichier nommé `transform.py`.

```
import boto3
import requests
from botocore.config import Config

# This function capitalizes all text in the original object
def lambda_handler(event, context):
    object_context = event["getObjectContext"]
    # Get the presigned URL to fetch the requested original object
    # from S3
    s3_url = object_context["inputS3Url"]
    # Extract the route and request token from the input context
    request_route = object_context["outputRoute"]
    request_token = object_context["outputToken"]

    # Get the original S3 object using the presigned URL
    response = requests.get(s3_url)
    original_object = response.content.decode("utf-8")

    # Transform all text in the original object to uppercase
    # You can replace it with your custom code based on your use case
    transformed_object = original_object.upper()

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3', config=Config(signature_version='s3v4'))
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)

    # Exit the Lambda function: return the status code
    return {'status_code': 200}
```

> **Note**
>
> L'exemple précédent de fonction Lambda charge l'ensemble de l'objet demandé en mémoire avant de le transformer et de le renvoyer au client. Vous pouvez autrement

diffuser l'objet à partir de S3 pour éviter de charger l'intégralité de l'objet dans la mémoire. Cette approche peut être utile lorsque vous travaillez avec des objets volumineux. Pour en savoir plus sur le streaming des réponses avec des points d'accès Object Lambda, consultez les exemples de streaming dans Utilisation de requêtes `GetObject` dans Lambda.

Lorsque vous écrivez une fonction Lambda à utiliser avec un point d'accès S3 Objet Lambda, la fonction est basée sur le contexte d'événement d'entrée que S3 Object Lambda fournit à la fonction Lambda. Le contexte de l'événement fournit des informations sur la demande effectuée dans l'événement transmis de S3 Object Lambda à Lambda. Il contient les paramètres que vous utiliserez pour créer la fonction Lambda.

Les champs utilisés pour créer la fonction Lambda précédente sont les suivants :

Le champ de `getObjectContext` indique les détails d'entrée et de sortie pour les connexions à Amazon S3 et à S3 Object Lambda. Il comporte les champs suivants :

- `inputS3Url` – Une URL présignée que la fonction Lambda peut utiliser pour télécharger l'objet original à partir du point d'accès de prise en charge. En utilisant une URL présignée, la fonction Lambda n'a pas besoin des autorisations de lecture Amazon S3 pour récupérer l'objet original et peut uniquement accéder à l'objet traité par chaque appel.
- `outputRoute` – Un jeton de routage qui est ajouté à l'URL S3 Object Lambda lorsque la fonction Lambda appelle `WriteGetObjectResponse` pour renvoyer l'objet transformé.
- `outputToken` – Un jeton utilisé par S3 Object Lambda pour correspondre à l'appel `WriteGetObjectResponse` avec l'appelant original lors du renvoi de l'objet transformé.

Pour en savoir plus sur tous les champs du contexte de l'événement, consultez Format et utilisation du contexte d'événement et Écriture de fonctions Lambda pour les points d'accès S3 Object Lambda.

3. Dans votre terminal local, saisissez la commande suivante pour installer le package `virtualenv` :

```
python -m pip install virtualenv
```

4. Dans votre terminal local, ouvrez le dossier `object-lambda` que vous avez créé précédemment, puis saisissez la commande suivante pour créer et initialiser un environnement virtuel nommé `venv`.

```
python -m virtualenv venv
```

5. Pour activer l'environnement virtuel, saisissez la commande suivante pour exécuter le fichier `activate` à partir du dossier de l'environnement :

   Pour les utilisateurs macOS, exécutez cette commande :

```
source venv/bin/activate
```

   Pour les utilisateurs Windows, exécutez cette commande :

```
.\venv\Scripts\activate
```

   Votre invite de commande change maintenant pour afficher (venv), indiquant que l'environnement virtuel est actif.

6. Pour installer les bibliothèques requises, exécutez les commandes suivantes ligne par ligne dans l'`venv`environnement virtuel.

   Ces commandes installent des versions mises à jour des dépendances de votre `lambda_handler`fonction Lambda. Ces dépendances sont les AWS SDK for Python (Boto3) et le module des demandes.

```
pip3 install boto3
```

```
pip3 install requests
```

7. Pour désactiver l'environnement virtuel, exécutez la commande suivante :

```
deactivate
```

8. Pour créer un package de déploiement avec les bibliothèques installées en tant que fichier `.zip` nommé `lambda.zip` à la racine du répertoire `object-lambda`, exécutez la commande les commandes ligne par ligne dans votre terminal local.

> **Tip**
>
> Les commandes suivantes peuvent nécessiter des ajustements pour fonctionner dans votre environnement particulier. Par exemple, une bibliothèque peut s'afficher dans `site-packages` ou dans `dist-packages`, et le premier dossier peut être `lib` ou `lib64`. En outre, il est possible de nommer le fichier `python` avec une version de Python différente. Vous pouvez utiliser la commande `pip show` pour localiser un package spécifique.

Pour les utilisateurs macOS, exécutez les commandes suivantes :

```
cd venv/lib/python3.8/site-packages
```

```
zip -r ../../../../lambda.zip .
```

Pour les utilisateurs Windows, exécutez les commandes suivantes :

```
cd .\venv\Lib\site-packages\
```

```
powershell Compress-Archive * ../../../lambda.zip
```

La dernière commande enregistre le package de déploiement à la racine du répertoire `object-lambda`.

9. Ajoutez le fichier de code de fonction `transform.py` à la racine de votre package de déploiement.

   Pour les utilisateurs macOS, exécutez les commandes suivantes :

   ```
   cd ../../../../
   ```

   ```
   zip -g lambda.zip transform.py
   ```

   Pour les utilisateurs Windows, exécutez les commandes suivantes :

```
cd ..\..\..\
```

```
powershell Compress-Archive -update transform.py lambda.zip
```

Lorsque vous aurez achevé cette étape, vous disposerez de la structure de répertoire suivante :

```
lambda.zip$
  # transform.py
  # __pycache__
  | boto3/
  # certifi/
  # pip/
  # requests/
  ...
```

Créer une fonction Lambda à l'aide d'un rôle d'exécution (console)

1. Connectez-vous à la AWS Lambda console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/lambda/.

2. Dans le panneau de navigation de gauche, choisissez Fonctions.
3. Choisissez Créer une fonction.
4. Choisissez Créer à partir de zéro.
5. Sous Informations de base, procédez comme suit :

   a. Sous Nom de la fonction, saisissez **`tutorial-object-lambda-function`**.

   b. Pour Exécution, choisissez Python 3.8 ou version ultérieure.

6. Développez la section Changer le rôle d'exécution par défaut. Sous Rôle d'exécution, choisissez Créer un nouveau rôle avec les autorisations Lambda de base.

   À l'étape 5 plus loin dans ce didacticiel, vous associerez AmazonS3 ObjectLambdaExecutionRolePolicy au rôle d'exécution de cette fonction Lambda.

7. Conservez les paramètres restants définis sur les valeurs par défaut.
8. Choisissez Créer une fonction.

Déployez le code de votre fonction Lambda avec les archives du fichier .zip et configurez la fonction Lambda (console)

1. Dans la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/, choisissez Functions dans le volet de navigation de gauche.
2. Choisissez la fonction Lambda que vous avez créée précédemment (par exemple, **`tutorial-object-lambda-function`**).
3. Dans la page des détails de la fonction Lambda, choisissez l'onglet Code. Dans la section Source du code, choisissez Charger à partir de, puis fichier .zip.
4. Choisissez Charger pour sélectionner votre fichier `.zip` local.
5. Choisissez le fichier `lambda.zip` que vous avez créé précédemment, puis choisissez Ouvrir.
6. Choisissez Enregistrer.
7. Dans la section Paramètres d'exécution, choisissez Modifier.
8. Dans la page Modifier les paramètres d'exécution, confirmez que Exécution est définie sur Python 3.8 ou version ultérieure.
9. Pour indiquer à l'exécution Lambda la méthode de gestionnaire dans votre code de fonction Lambda à appeler, saisissez **`transform.lambda_handler`** pour Gestionnaire.

   Si vous configurez une fonction dans Python, la valeur du paramètre du gestionnaire sera le nom du fichier et le nom du module du gestionnaire, séparés par un point. Par exemple, `transform.lambda_handler` appelle la méthode `lambda_handler` définie dans le fichier `transform.py`.
10. Choisissez Enregistrer.
11. (Facultatif) Dans la page des détails de votre fonction Lambda, choisissez l'onglet Configuration. Dans le panneau de navigation de gauche, choisissez Configuration générale, puis Modifier. Dans le champ Délai d'expiration, saisissez **`1`** min **`0`** s. Conservez les paramètres restants définis sur les valeurs par défaut, puis choisissez Enregistrer.

    Le Délai d'expiration est la durée que Lambda autorise pour l'exécution d'une fonction pour appel avant de l'arrêter. Le durée par défaut est de 3 secondes. La durée maximale d'une fonction Lambda utilisée par S3 Object Lambda est de 60 secondes. La tarification est basée sur la quantité de mémoire configurée et la durée pendant laquelle votre code s'exécute.

Étape 5 : Configurer une politique IAM pour le rôle d'exécution de votre fonction Lambda

Pour permettre à votre fonction Lambda de fournir des données personnalisées et des en-têtes de réponse à l'appelant `GetObject`, le rôle d'exécution de votre fonction Lambda doit disposer des autorisations IAM pour appeler l'API `WriteGetObjectResponse`.

Pour attacher une politique IAM au rôle de votre fonction Lambda

1. Dans la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/, choisissez Functions dans le volet de navigation de gauche.
2. Choisissez la fonction que vous avez créée à l'étape 4 (par exemple, **`tutorial-object-lambda-function`**).
3. Dans la page des détails de votre fonction Lambda, choisissez l'onglet Configuration, puis Autorisations dans le panneau de navigation de gauche.
4. Sous Rôle d'exécution, cliquez sur le lien du Nom de rôle. La console IAM s'ouvre.
5. Sur la page Summary (Résumé) de la console IAM pour le rôle d'exécution de votre fonction Lambda, choisissez l'onglet Permissions (Autorisations). Ensuite, dans le menu Add Permissions (Ajouter des autorisations), choisissez Attach policies (Attacher des politiques).
6. Dans la page Attacher des autorisations, saisissez **`AmazonS3ObjectLambdaExecutionRolePolicy`** dans le champ de recherche pour filtrer la liste des politiques Cochez la case à côté du nom de la politique AmazonS3 ObjectLambdaExecutionRolePolicy.
7. Choisissez Attach Policies (Attacher des politiques).

Étape 6 : Créer un point d'accès S3 Object Lambda

Un point d'accès S3 Object Lambda offre la flexibilité d'appeler une fonction Lambda directement à partir d'une demande GET S3 afin que la fonction puisse traiter les données récupérées à partir d'un point d'accès S3. Lors de la création et de la configuration d'un point d'accès S3 Object Lambda, vous devez indiquer la fonction Lambda à appeler et fournir le contexte de l'événement au format JSON en tant que paramètres personnalisés à utiliser par Lambda.

Créer un point d'accès S3 Object Lambda

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Dans la page Points d'accès Object Lambda), choisissez Créer un point d'accès Object Lambda).
4. Pour Nom du point d'accès Lambda d'objet, saisissez le nom que vous souhaitez utiliser pour le point d'accès Object Lambda (par exemple, **`tutorial-object-lambda-accesspoint`**).
5. Pour Point d'accès de prise en charge, saisissez ou accédez au point d'accès standard que vous avez créé à l'étape 3 (par exemple, **`tutorial-access-point`**), puis choisissez Choisir un point d'accès de prise en charge.
6. Pour les API S3, pour récupérer des objets du compartiment S3 afin que la fonction Lambda les traite, sélectionnez. GetObject
7. Pour appeler une fonction Lambda, vous pouvez choisir l'une des deux options suivantes de ce didacticiel.
   - Choisissez Choisissez parmi les fonctions dans votre compte, puis choisissez la fonction Lambda que vous avez créée à l'étape 4(par exemple, **`tutorial-object-lambda-function`**) dans la liste déroulante Fonction Lambda.
   - Choisissez Saisir l'ARN, puis entrez l'Amazon Resource Name (ARN) de la fonction Lambda que vous avez créée à l'étape 4
8. Pour Version de la fonction Lambda, choisissez $LATEST (la dernière version de la fonction Lambda que vous avez créée à l'étape 4).
9. (Facultatif) Si vous avez besoin de votre fonction Lambda pour reconnaître et traiter les demandes GET avec des en-têtes de plage et de numéro de pièce, sélectionnez La fonction Lambda prend en charge les demandes utilisant une plage et La fonction Lambda prend en charge les demandes utilisant des numéros de pièce. Sinon, décochez ces deux cases.

   Pour en savoir plus sur l'utilisation des plages ou des numéros de pièce avec S3 Object Lambda, consultez Utilisation des en-têtes Range et partNumber.
10. (Facultatif) Sous Charge utile – facultatif, ajoutez un texte JSON pour fournir des informations supplémentaires à votre fonction Lambda.

    Une charge utile est un texte JSON facultatif que vous pouvez fournir à votre fonction Lambda comme entrée pour tous les appels provenant d'un point d'accès S3 Object Lambda spécifique.

Pour personnaliser les comportements de plusieurs points d'accès Object Lambda qui appellent la même fonction Lambda, vous pouvez configurer des charges utiles avec différents paramètres, augmentant ainsi la flexibilité de votre fonction Lambda.

Pour en savoir plus sur les charges utiles, consultez Format et utilisation du contexte d'événement.

11. (Facultatif) Pour Métriques de demande – facultatif, choisissez Activer ou Désactiver pour ajouter la surveillance Amazon S3 à votre point d'accès Object Lambda. Les statistiques relatives aux demandes sont facturées au CloudWatch tarif standard d'Amazon. Pour en savoir plus, consultez CloudWatch Tarification.
12. Sous Politique de point d'accès Object Lambda – facultatif, conservez le paramètre par défaut.

    (Facultatif) Vous pouvez définir une politique de ressource. Cette politique de ressource permet à l'autorisation d'API `GetObject` d'utiliser le point d'accès Object Lambda spécifié.
13. Conservez les paramètres restants définis sur les valeurs par défaut, puis choisissez Créer un point d'accès Object Lambda.

Étape 7 : Afficher les données transformées

S3 Object Lambda est maintenant prêt à transformer vos données pour votre cas d'utilisation. Dans ce didacticiel, S3 Object Lambda transforme tout le texte de votre objet en majuscules.

Sous-étapes

- Afficher les données transformées dans votre point d'accès S3 Object Lambda
- Exécuter un script Python pour imprimer les données originales et transformées

Afficher les données transformées dans votre point d'accès S3 Object Lambda

Lorsque vous demandez de récupérer un fichier via votre point d'accès S3 Object Lambda, vous effectuez un appel d'API `GetObject` à S3 Object Lambda. S3 Object Lambda appelle la fonction Lambda pour transformer vos données, puis retourne les données transformées en réponse à l'appel d'API `GetObject` S3 standard.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.

3. Sur la page Points d'accès Lambda d'objet, choisissez le point d'accès S3 Object Lambda que vous avez créé à l'étape 6 (par exemple, **`tutorial-object-lambda-accesspoint`**).
4. Dans l'onglet Objets de votre point d'accès S3 Object Lambda, sélectionnez le fichier portant le même nom (par exemple, `tutorial.txt`) comme celui que vous avez chargé dans le compartiment S3 à l'étape 2.

   Ce fichier doit contenir toutes les données transformées.
5. Pour afficher les données transformées, choisissez Ouvrir ou Télécharger.

Exécuter un script Python pour imprimer les données originales et transformées

Vous pouvez utiliser S3 Object Lambda avec vos applications existantes. Pour ce faire, mettez à jour la configuration de votre application pour utiliser le nouvel ARN du point d'accès S3 Object Lambda que vous avez créé à l'étape 6 afin de récupérer des données depuis S3.

L'exemple de script Python suivant imprime à la fois les données originales à partir du compartiment S3 et les données transformées à partir du point d'accès S3 Object Lambda.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Sur la page Points d'accès Lambda d'objet, choisissez le bouton radio situé à gauche du point d'accès S3 Object Lambda que vous avez créé à l'étape 6 (par exemple, **`tutorial-object-lambda-accesspoint`**).
4. Choisissez Copier l'ARN.
5. Enregistrez l'ARN pour utilisation ultérieure.
6. Écrivez un script Python sur votre machine locale pour imprimer les données originales (par exemple, `tutorial.txt`) à partir de votre compartiment S3 et les données transformées (par exemple, `tutorial.txt`) à partir de votre point d'accès S3 Object Lambda. Vous pouvez utiliser l'exemple de script suivant.

```
import boto3
from botocore.config import Config

s3 = boto3.client('s3', config=Config(signature_version='s3v4'))

def getObject(bucket, key):
```

```
    objectBody = s3.get_object(Bucket = bucket, Key = key)
    print(objectBody["Body"].read().decode("utf-8"))
    print("\n")

print('Original object from the S3 bucket:')
# Replace the two input parameters of getObject() below with
# the S3 bucket name that you created in Step 1 and
# the name of the file that you uploaded to the S3 bucket in Step 2
getObject("tutorial-bucket",
          "tutorial.txt")

print('Object transformed by S3 Object Lambda:')
# Replace the two input parameters of getObject() below with
# the ARN of your S3 Object Lambda Access Point that you saved earlier and
# the name of the file with the transformed data (which in this case is
# the same as the name of the file that you uploaded to the S3 bucket
# in Step 2)
getObject("arn:aws:s3-object-lambda:us-west-2:111122223333:accesspoint/tutorial-
object-lambda-accesspoint",
          "tutorial.txt")
```

7. Enregistrez votre script Python avec un nom personnalisé (par exemple, `tutorial_print.py`) dans le dossier (par exemple, `object-lambda`) que vous avez créé à l'étape 4 sur votre machine locale.

8. Dans votre terminal local, exécutez la commande suivante à partir de la racine du répertoire (par exemple, `object-lambda`) que vous avez créé à l'étape 4.

```
python3 tutorial_print.py
```

Vous devriez voir à la fois les données originales et les données transformées (tout le texte en majuscules) sur le terminal. Par exemple, vous devriez voir quelque chose ressemblant au texte suivant.

```
Original object from the S3 bucket:
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
returning it to an application.

Object transformed by S3 Object Lambda:
AMAZON S3 OBJECT LAMBDA TUTORIAL:
YOU CAN ADD YOUR OWN CODE TO PROCESS DATA RETRIEVED FROM S3 BEFORE
```

```
RETURNING IT TO AN APPLICATION.
```

Étape 8 : Nettoyer

Si vous avez transformé vos données via S3 Object Lambda comme un simple exercice d'apprentissage, supprimez les ressources AWS que vous avez allouées afin de ne plus accumuler de frais.

Sous-étapes

- Supprimer le point d'accès Object Lambda
- Supprimer le point d'accès S3
- Supprimer le rôle d'exécution de votre fonction Lambda.
- Supprimer la fonction Lambda
- Supprimer le groupe de CloudWatch journaux
- Supprimer le fichier original dans le compartiment source S3
- Supprimer le compartiment source S3
- Supprimer l'utilisateur IAM

Supprimer le point d'accès Object Lambda

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Sur la page Points d'accès Lambda d'objet, choisissez le bouton radio situé à gauche du point d'accès S3 Object Lambda que vous avez créé à l'étape 6 (par exemple, **`tutorial-object-lambda-accesspoint`**).
4. Choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer votre point d'accès Object Lambda en saisissant son nom dans le champ de texte qui s'affiche, puis choisissez Supprimer.

Supprimer le point d'accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Points d'accès.
3. Accédez au point d'accès que vous avez créé à l'étape 3 (par exemple, **`tutorial-access-point`**), puis choisissez le bouton radio en regard du nom du point d'accès.
4. Choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer votre point d'accès Object Lambda en saisissant son nom dans le champ de texte qui s'affiche, puis choisissez Supprimer.

Supprimer le rôle d'exécution de votre fonction Lambda.

1. Connectez-vous à la AWS Lambda console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/lambda/.
2. Dans le panneau de navigation de gauche, choisissez Fonctions.
3. Choisissez la fonction que vous avez créée à l'étape 4 (par exemple, **`tutorial-object-lambda-function`**).
4. Dans la page des détails de votre fonction Lambda, choisissez l'onglet Configuration, puis Autorisations dans le panneau de navigation de gauche.
5. Sous Rôle d'exécution, cliquez sur le lien du Nom de rôle. La console IAM s'ouvre.
6. Dans la page Récapitulatif de la console IAM du rôle d'exécution de votre fonction Lambda, choisissez Supprimer le rôle.
7. Dans la boîte de dialogue Supprimer le rôle, choisissez Oui, supprimer.

Supprimer la fonction Lambda

1. Dans la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/, choisissez Functions dans le volet de navigation de gauche.
2. Cochez la case à gauche du nom de la fonction que vous avez créée à l'étape 4 (par exemple, **`tutorial-object-lambda-function`**).
3. Choisissez Actions, puis Supprimer.
4. Dans la boîte de dialogue Supprimer une fonction, choisissez Supprimer.

Supprimer le groupe de CloudWatch journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation de gauche, choisissez Groupes de journaux.
3. Recherchez le groupe de journaux dont le nom se termine par la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **`tutorial-object-lambda-function`**).
4. Cochez la case située à gauche du nom du groupe de journaux.
5. Choisissez Actions, puis Supprimer le ou les groupes de journaux.
6. Dans la boîte de dialogue Supprimer le ou les groupes de journaux, choisissez Supprimer.

Supprimer le fichier original dans le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Nom du compartiment, choisissez le nom du compartiment vers lequel vous avez chargé le fichier original à l'étape 2 (par exemple, **`tutorial-bucket`**).
4. Cochez la case située à gauche du nom de l'objet que vous souhaitez supprimer (par exemple, `tutorial.txt`).
5. Choisissez Supprimer.
6. Dans la page Supprimer des objets, dans la section Supprimer définitivement les objets ?, confirmez que vous souhaitez supprimer cet objet en saisissant **`permanently delete`** dans la zone de texte.
7. Choisissez Supprimer les objets.

Supprimer le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le bouton radio en regard du nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**).
4. Choisissez Supprimer.
5. Dans la page Supprimer le compartiment, confirmez que vous souhaitez supprimer le compartiment en saisissant le nom de ce dernier dans le champ de texte, puis choisissez Supprimer le compartiment.

Supprimer l'utilisateur IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Utilisateurs, puis cochez la case en regard du nom de l'utilisateur à supprimer.
3. En haut de la page, choisissez Supprimer.
4. Dans la boîte de dialogue Supprimer ***`un nom d'utilisateur`*** ?, saisissez le nom de l'utilisateur dans le champ de saisie de texte afin de confirmer la suppression de l'utilisateur. Choisissez Supprimer.

Étapes suivantes

Après avoir terminé ce didacticiel, vous pourrez personnaliser la fonction Lambda pour votre cas d'utilisation afin de modifier les données renvoyées par des demandes GET S3 standard.

Voici une liste de cas d'utilisation courants pour S3 Object Lambda :

- Masquer les données sensibles à des fins de sécurité et de conformité.

  Pour plus d'informations, consultez Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend.
- Filtrer certaines lignes de données pour fournir des informations précises.
- Enrichir les données avec des informations provenant d'autres services ou bases de données.
- Conversion entre les formats de données, comme la conversion XML en JSON pour la compatibilité des applications.
- Compression ou décompression des fichiers pendant qu'ils sont téléchargés.
- Redimensionnement des images et insertion d'un filigrane.

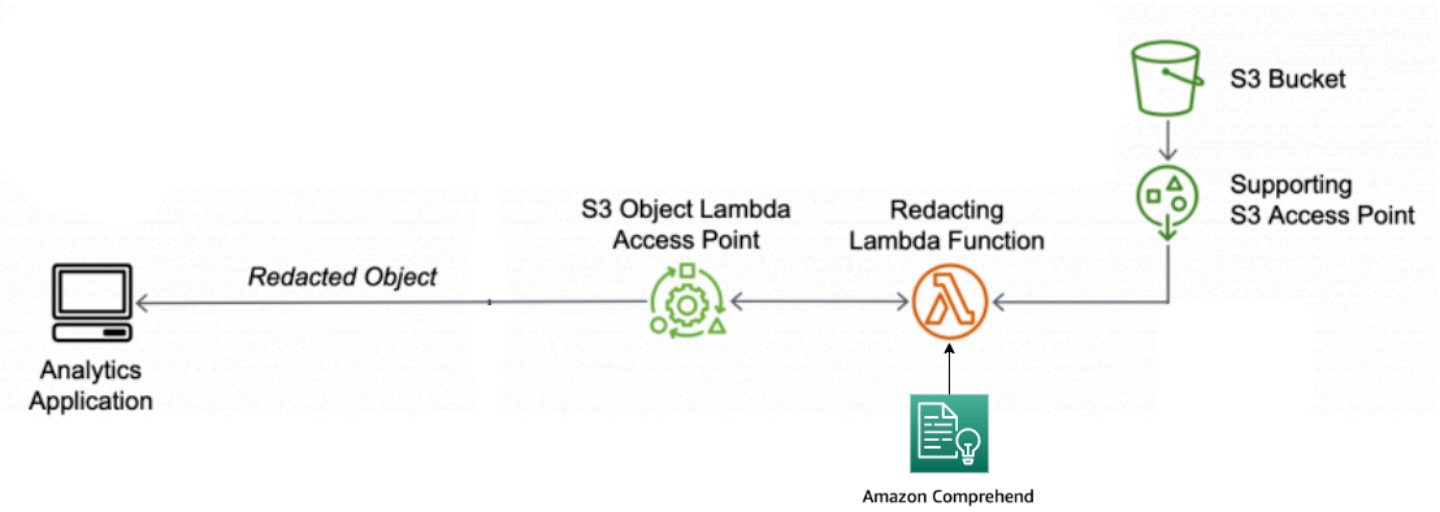
  Pour plus d'informations, consultez Didacticiel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération.
- Mise en œuvre de règles d'autorisation personnalisées pour accéder aux données.

Pour en savoir plus sur S3 Object Lambda, consultez Transformation d'objets avec S3 Object Lambda.

Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend

Lorsque vous utilisez Amazon S3 pour des jeux de données partagés auxquels plusieurs applications et utilisateurs peuvent accéder, il est important de restreindre les informations privilégiées, telles que les données personnelles identifiables (DPI), aux entités autorisées uniquement. Par exemple, lorsqu'une application marketing utilise des données contenant des DPI, elle doit d'abord masquer les DPI pour répondre aux exigences de confidentialité des données. En outre, lorsqu'une application d'analyse utilise un jeu de données d'inventaire d'ordre de production, elle peut devoir d'abord effacer les informations de carte de crédit du client afin d'éviter toute fuite de données involontaire.

Grâce à S3 Object Lambda et une fonction AWS Lambda optimisée par Amazon Comprehend, vous pouvez protéger les DPI récupérées à partir de S3 avant de les renvoyer à une application. Plus précisément, vous pouvez utiliser la fonction Lambda préconstruite en tant que fonction d'expurgation et l'attacher à un point d'accès S3 Object Lambda. Lorsqu'une application (par exemple, une application d'analyse) envoie des demandes GET standard S3, ces demandes effectuées via le point d'accès S3 Object Lambda appellent la fonction de suppression Lambda préconstruite afin de détecter et d'expurger les DPI récupérées à partir d'un compartiment S3 via un point d'accès S3 pris en charge. Ensuite, le point d'accès S3 Object Lambda retourne le résultat expurgé à l'application.



Dans le processus, la fonction Lambda préconstruite utilise Amazon Comprehend, un service de traitement du langage naturel (PNL), qui permet de saisir les variations dans la façon dont les DPI sont représentées, quelle que soit la façon dont les DPI existent dans le texte (par exemple, numériquement ou sous forme de combinaison de mots et de chiffres). Amazon Comprehend peut même utiliser le contexte dans le texte pour comprendre si un numéro à 4 chiffres est un code PIN, les quatre derniers numéros d'un numéro de sécurité sociale (NSS) ou une année.

Amazon Comprehend traite n'importe quel fichier texte au format UTF-8 et peut protéger les DPI à grande échelle sans affecter la précision. Pour en savoir plus, consultez Qu'est-ce qu'Amazon Comprehend ? dans le guide du développeur Amazon Comprehend.

Objectif

Dans ce didacticiel, vous apprendrez à utiliser S3 Object Lambda avec la fonction Lambda préconstruite `ComprehendPiiRedactionS3ObjectLambda`. Cette fonction utilise Amazon Comprehend pour détecter les entités DPI. Elle supprime ensuite ces entités en les remplaçant par des astérisques. En expurgeant des DPI, vous masquez les données sensibles, ce qui peut aider en matière de sécurité et de conformité.

Vous apprendrez également à utiliser et à configurer une AWS Lambda fonction prédéfinie dans le AWS Serverless Application Repositorypour qu'elle fonctionne avec S3 Object Lambda afin de faciliter le déploiement.

Rubriques

Conditions préalables : Créer un utilisateur IAM avec des autorisations

Avant de commencer ce didacticiel, vous devez disposer d'un AWS compte auquel vous pouvez vous connecter en tant qu' AWS Identity and Access Management utilisateur (utilisateur IAM) avec les autorisations appropriées.

Vous pouvez créer un utilisateur IAM pour le didacticiel. Pour terminer ce didacticiel, votre utilisateur IAM doit joindre les politiques IAM suivantes pour accéder aux AWS ressources pertinentes et effectuer des actions spécifiques.

> **Note**
>
> Pour des raisons de simplicité, ce didacticiel crée et utilise un utilisateur IAM. Après avoir terminé ce didacticiel, n'oubliez pas de Supprimer l'utilisateur IAM. Pour une utilisation en production, nous vous recommandons de suivre les Meilleures pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM. Une bonne pratique requiert que les utilisateurs humains utilisent une fédération avec un fournisseur d'identité pour accéder à AWS avec des informations d'identification temporaires. Une autre bonne pratique consiste à exiger que les charges de travail utilisent des informations d'identification temporaires avec des rôles IAM pour accéder à AWS. Pour en savoir plus sur l'utilisation AWS IAM Identity Center pour créer des utilisateurs avec des informations d'identification temporaires, voir Getting started dans le guide de AWS IAM Identity Center l'utilisateur.
>
> Pour plus de simplicité, ce didacticiel utilise des stratégies d'accès complet. Pour utilisation en production, nous vous recommandons d'accorder uniquement les autorisations minimales nécessaires à votre cas d'utilisation, conformément aux Bonnes pratiques de sécurité.

Votre utilisateur IAM a besoin des politiques AWS gérées suivantes :

- AmazonS3 FullAccess — Accorde des autorisations pour toutes les actions Amazon S3, y compris les autorisations pour créer et utiliser un point d'accès Object Lambda.
- AWSLambda_FullAccess— Accorde des autorisations à toutes les actions Lambda.
- AWSCloudFormationFullAccess— Accorde des autorisations pour toutes les AWS CloudFormation actions.
- IAM FullAccess — Accorde des autorisations à toutes les actions IAM.
- IAM AccessAnalyzerReadOnlyAccess — Accorde l'autorisation de lire toutes les informations d'accès fournies par IAM Access Analyzer.

Vous pouvez attacher directement ces politiques existantes lors de la création d'un utilisateur IAM. Pour en savoir plus sur la création d'un utilisateur IAM, consultez Créer votre premier groupe d'utilisateurs et d'administrateurs IAM dans le guide de l'utilisateur IAM.

En outre, votre utilisateur IAM nécessite une politique gérée par le client. Pour accorder à l'utilisateur IAM des autorisations sur toutes les AWS Serverless Application Repository ressources et actions, vous devez créer une stratégie IAM et l'associer à l'utilisateur IAM.

Créer et attacher une politique IAM à votre utilisateur IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Choisissez Créer une politique.
4. Dans l'onglet Éditeur visuel, pour Service, choisissez .Choisir un service. Ensuite, choisissez Serverless Application Repository.
5. Pour Actions, sous Actions manuelles, sélectionnez Toutes les actions du Serverless Application Repository (serverlessrepo :*) pour ce didacticiel.

   En tant que bonne pratique de sécurité, vous devez autoriser un utilisateur à accéder uniquement aux actions et ressources qui lui sont nécessaires, selon votre cas d'utilisation. Pour en savoir plus, consultez Bonnes pratiques de sécurité dans IAM que vous trouverez dans le guide de l'utilisateur IAM.
6. Pour Ressources, choisissez Toutes les ressources pour ce didacticiel.

   La bonne pratique consiste à définir des autorisations pour des ressources spécifiques uniquement dans des comptes spécifiques. Vous pouvez autrement accorder le moindre privilège en utilisant des clés de condition. Pour en savoir plus, consultez Accorder le moindre privilège dans le guide de l'utilisateur IAM.
7. Choisissez Suivant : Balises.
8. Choisissez Suivant : Examiner.
9. Dans la page Examiner une politique, saisissez un nom (par exemple, **`tutorial-serverless-application-repository`**) et une Description (facultatif) pour la politique que vous êtes en train de créer. Examinez le récapitulatif de la politique afin de vérifier que vous avez accordé les autorisations souhaitées, puis choisissez Créer une politique pour enregistrer votre nouvelle politique.
10. Dans le volet de navigation de gauche, choisissez Utilisateurs. Ensuite, choisissez l'utilisateur IAM pour ce didacticiel.
11. Dans la page Récapitulatif de l'utilisateur sélectionné, choisissez l'onglet Autorisations, puis choisissez Ajouter des autorisations.
12. Sous Octroyer des autorisations, choisissez Attacher directement les politiques existantes.
13. Cochez la case en regard de la politique que vous avez créée (par exemple, **`tutorial-serverless-application-repository`**), puis choisissez Suivant : Examiner.

14. Sous Récapitulatif des autorisations, examinez le récapitulatif de la politique afin de vérifier que vous avez attaché la politique prévue. Choisissez ensuite Ajouter des autorisations.

Étape 1 : Créer un compartiment S3

Créez un compartiment pour stocker les données originales que vous prévoyez de transformer.

Pour créer un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
4. Pour Nom du compartiment, saisissez un nom (par exemple, **`tutorial-bucket`**) pour votre compartiment.

   Pour en savoir plus sur les règles d'attribution de noms des compartiments Amazon S3, consultez Règles de dénomination de compartiment.
5. Dans Région, choisissez la Région AWS dans laquelle le compartiment doit résider.

   Pour en savoir plus sur les régions des compartiments, consultez Présentation des compartiments.
6. Pour Paramètres de blocage de l'accès public à ce compartiment, conservez les paramètres par défaut (Bloquer tout accès public est activé).

   Nous vous recommandons de laisser tous les paramètres de blocage de l'accès public activés, sauf si vous devez en désactiver un ou plusieurs pour votre cas d'utilisation. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.
7. Pour les paramètres restants, conservez les paramètres par défaut.

   (Facultatif) Si vous souhaitez configurer des paramètres de compartiment supplémentaires pour votre cas d'utilisation particulier, consultez Créer un compartiment.
8. Choisissez Créer un compartiment.

Étape 2 : Charger un fichier dans le compartiment S3

Chargez un fichier texte contenant des DPI connues de différents types, comme des noms, des informations bancaires, des numéros de téléphone et des NSS, dans le compartiment S3 en tant que données originales que vous supprimerez des DPI plus loin dans ce didacticiel.

Par exemple, vous pouvez charger le fichier `tutorial.txt` suivant. Voici un exemple de fichier d'entrée Amazon Comprehend.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services,
LLC credit card account 1111-0000-1111-0008 has a minimum payment
of $24.53 that is due by July 31st. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account number XXXXXX1111 with the routing number XXXXX0000.

Your latest statement was mailed to 100 Main Street, Any City,
WA 98121.
After your payment is received, you will receive a confirmation
text message at 206-555-0100.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at 206-555-0199 or
email at support@anycompany.com.
```

Charger un fichier dans un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**) pour y charger votre fichier.
4. Sous l'onglet Objets de votre compartiment, choisissez Charger.
5. Dans la page Charger, sous Fichiers et dossiers, choisissez Ajouter des fichiers.
6. Choisissez un fichier à charger, puis choisissez Ouvrir. Par exemple, vous pouvez charger le fichier `tutorial.txt` mentionné précédemment.
7. Choisissez Charger.

Étape 3 : Créer un point d'accès S3

Pour utiliser un point d'accès S3 Object Lambda afin d'accéder aux données originales et de les transformer, vous devez créer un point d'accès S3 et l'associer au compartiment S3 que vous avez créé à l'étape 1. Le point d'accès doit se trouver dans le même Région AWS emplacement que les objets que vous souhaitez transformer.

Plus loin dans ce didacticiel, vous utiliserez ce point d'accès comme point d'accès de prise en charge pour votre point d'accès Object Lambda.

Créer un point d'accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation, choisissez Points d'accès.
3. Dans la page Points d'accès, choisissez Créer un point d'accès.
4. Dans le champ Nom du point d'accès, saisissez le nom (par exemple, **`tutorial-pii-access-point`**) du point d'accès.

   Pour en savoir plus sur l'attribution de noms aux points d'accès, consultez Règles relatives à l'attribution de noms pour les points d'accès Amazon S3.
5. Dans Nom du compartiment, saisissez le nom du compartiment que vous avez créé à l'étape 1(par exemple, **`tutorial-bucket`**). S3 attache le point d'accès à ce compartiment.

   (Facultatif) Vous pouvez choisir Parcourir S3 pour parcourir et rechercher les compartiments inclus dans votre compte. Si vous choisissez Parcourir S3, choisissez le compartiment souhaité, puis choisissez Choisir un chemin pour renseigner le champ Nom du compartiment avec le nom de ce compartiment.
6. Pour Origine du réseau, choisissez Internet.

   Pour en savoir plus sur les origines de réseau des points d'accès, consultez Création de points d'accès restreints à un virtual private cloud.
7. Par défaut, tous les paramètres de blocage d'accès public sont activés pour votre point d'accès. Nous vous recommandons de conserver l'option Bloquer tous les accès publics activée. Pour en savoir plus, consultez Gestion de l'accès public aux points d'accès.
8. Pour tous les autres paramètres de point d'accès, conservez les paramètres par défaut.

(Facultatif) Vous pouvez modifier les paramètres du point d'accès afin de prendre en charge votre cas d'utilisation. Pour ce didacticiel, nous vous recommandons de conserver les paramètres par défaut.

(Facultatif) Si vous devez gérer l'accès à votre point d'accès, vous pouvez indiquer une politique de point d'accès. Pour plus d'informations, consultez Exemples de stratégie de point d'accès.

9. Choisissez Créer un point d'accès.

Étape 4 : Configurer et déployer une fonction Lambda préconstruite

Pour effacer les DPI, configurez et déployez la fonction AWS Lambda `ComprehendPiiRedactionS3ObjectLambda` préconstruite à utiliser avec votre point d'accès S3 Object Lambda.

Configurez et déployez la fonction Lambda

1. Connectez-vous au AWS Management Console et visualisez la `ComprehendPiiRedactionS3ObjectLambda`fonction dans le AWS Serverless Application Repository.
2. Pour les Paramètres de l'application, sous Nom de l'application, conservez la valeur par défaut (`ComprehendPiiRedactionS3ObjectLambda`) pour ce didacticiel.

   (Facultatif) Vous pouvez saisir le nom que vous souhaitez donner à cette application. Vous pouvez faire ceci si vous prévoyez de configurer plusieurs fonctions Lambda pour différents besoins d'accès au même jeu de données partagé.
3. Pour MaskCharacter, conservez la valeur par défaut (*****). Le caractère de masque remplace chaque caractère de l'entité DPI expurgée.
4. Pour MaskMode, conservez la valeur par défaut (MASK). La MaskModevaleur indique si l'entité PII est expurgée avec le `MASK` caractère ou la `PII_ENTITY_TYPE` valeur.
5. Pour supprimer les types de données spécifiés, pour PiiEntityTypes, conservez la valeur par défaut ALL. La PiiEntityTypesvaleur indique les types d'entités PII à prendre en compte pour la rédaction.

   Pour en savoir plus sur la liste des types d'entités DPI pris en charge, consultez Détecter les données personnelles identifiables (DPI) dans le guide du développeur Amazon Comprehend.
6. Conservez les paramètres restants définis sur les valeurs par défaut.

(Facultatif) Si vous souhaitez configurer des paramètres supplémentaires pour votre cas d'utilisation spécifique, consultez la section Fichier Readme sur le côté gauche de la page.

7. Cochez la case en regard de Je reconnais que cette application crée des rôles IAM personnalisés.
8. Choisissez Déployer).
9. Dans la page de la nouvelle application, sous Ressources, choisissez l'ID logique de la fonction Lambda que vous avez déployée pour examiner la fonction dans la page de la fonction Lambda.

Étape 5 : Créer un point d'accès S3 Object Lambda

Un point d'accès S3 Object Lambda offre la flexibilité d'appeler une fonction Lambda directement à partir d'une demande GET S3 afin que la fonction puisse supprimer les DPI récupérées à partir d'un point d'accès S3. Lors de la création et de la configuration d'un point d'accès S3 Object Lambda, vous devez indiquer la fonction d'expurgation Lambda à appeler et fournir le contexte d'événement au format JSON en tant que paramètres personnalisés pour utilisation par Lambda.

Le contexte d'événement fournit des informations sur la demande effectuée dans l'événement transmis de S3 Object Lambda à Lambda. Pour en savoir plus sur tous les champs du contexte d'événement, consultez Format et utilisation du contexte d'événement.

Créer un point d'accès S3 Object Lambda

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Dans la page Points d'accès Object Lambda), choisissez Créer un point d'accès Object Lambda).
4. Pour Nom du point d'accès Lambda d'objet, saisissez le nom que vous souhaitez utiliser pour le point d'accès Object Lambda (par exemple, **`tutorial-pii-object-lambda-accesspoint`**).
5. Pour Point d'accès de prise en charge, saisissez ou accédez au point d'accès standard que vous avez créé à l'étape 3 (par exemple, **`tutorial-pii-access-point`**), puis choisissez Choisir un point d'accès de prise en charge.
6. Pour les API S3, pour récupérer des objets du compartiment S3 afin que la fonction Lambda les traite, sélectionnez. GetObject

7. Pour Appeler une fonction Lambda, vous pouvez choisir l'une des deux options suivantes pour ce didacticiel.

   - Choisissez Choisir parmi les fonctions de votre compte et choisissez la fonction Lambda que vous avez déployée à l'étape 4 (par exemple, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) dans la liste déroulante Fonction Lambda.
   - Choisissez Saisir l'ARN, puis saisissez l'Amazon Resource Name (ARN) de la fonction Lambda que vous avez créée à l'étape 4.

8. PourVersion de la fonction Lambda, choisissez $LATEST (la dernière version de la fonction Lambda que vous avez déployée à l'étape 4).

9. (Facultatif) Si vous avez besoin de votre fonction Lambda pour reconnaître et traiter les demandes GET avec des en-têtes de plage et de numéro de partie, sélectionnez La fonction Lambda prend en charge les demandes utilisant la plage et La fonction Lambda prend en charge les demandes utilisant des numéros de partie. Sinon, décochez ces deux cases.

   Pour en savoir plus sur l'utilisation des plages ou des numéros de pièce avec S3 Object Lambda, consultez Utilisation des en-têtes Range et partNumber.

10. (Facultatif) Sous Charge utile – facultatif, ajoutez un texte JSON pour fournir des informations supplémentaires à votre fonction Lambda.

    Une charge utile est un texte JSON facultatif que vous pouvez fournir à votre fonction Lambda comme entrée pour tous les appels provenant d'un point d'accès S3 Object Lambda spécifique. Pour personnaliser les comportements de plusieurs points d'accès Object Lambda qui appellent la même fonction Lambda, vous pouvez configurer des charges utiles avec différents paramètres, augmentant ainsi la flexibilité de votre fonction Lambda.

    Pour en savoir plus sur les charges utiles, consultez Format et utilisation du contexte d'événement.

11. (Facultatif) Pour Métriques de demande – facultatif, choisissez Activer ou Désactiver pour ajouter la surveillance Amazon S3 à votre point d'accès Object Lambda. Les statistiques relatives aux demandes sont facturées au CloudWatch tarif standard d'Amazon. Pour en savoir plus, consultez CloudWatch Tarification.

12. Sous Politique de point d'accès Object Lambda – facultatif, conservez le paramètre par défaut.

    (Facultatif) Vous pouvez définir une politique de ressource. Cette politique de ressource permet à l'autorisation d'API `GetObject` d'utiliser le point d'accès Object Lambda spécifié.

13. Conservez les paramètres restants définis sur les valeurs par défaut, et choisissez Créer un point d'accès Object Lambda.

Étape 6 : Utiliser le point d'accès S3 Object Lambda pour récupérer le fichier expurgé

Maintenant, S3 Object Lambda est prêt à expurger les DPI de votre fichier original.

Pour utiliser le point d'accès S3 Object Lambda pour récupérer le fichier expurgé

Lorsque vous demandez de récupérer un fichier via votre point d'accès S3 Object Lambda, vous effectuez un appel d'API `GetObject` à S3 Object Lambda. L'objet S3 Lambda appelle la fonction Lambda pour expurger vos DPI et retourne les données transformées comme réponse à l'appel d'API`GetObject` S3 standard.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Sur la page Points d'accès Lambda d'objet, choisissez le point d'accès S3 Object Lambda que vous avez créé à l'étape 5 (par exemple, **`tutorial-pii-object-lambda-accesspoint`**).
4. Dans l'onglet Objets de votre point d'accès S3 Object Lambda, sélectionnez le fichier portant le même nom (par exemple, `tutorial.txt`) comme celui que vous avez chargé dans le compartiment S3 à l'étape 2.

   Ce fichier doit contenir toutes les données transformées.
5. Pour afficher les données transformées, choisissez Ouvrir ou Télécharger.

   Vous devriez pouvoir voir le fichier expurgé, comme illustré dans l'exemple suivant.

```
Hello *********. Your AnyCompany Financial Services,
LLC credit card account ******************** has a minimum payment
of $24.53 that is due by *********. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account ********** with the routing number *********.

Your latest statement was mailed to ********************************.
After your payment is received, you will receive a confirmation
text message at ************.
```

```
If you have questions about your bill, AnyCompany Customer Service
is available by phone at ************ or
email at **********************.
```

Étape 7 : nettoyer

Si vous avez expurgé vos données via S3 Object Lambda uniquement à des fins d'apprentissage, supprimez AWS les ressources que vous avez allouées afin de ne plus payer de frais.

Sous-étapes

- Supprimer le point d'accès Object Lambda
- Supprimer le point d'accès S3
- Supprimer la fonction Lambda
- Supprimer le groupe de CloudWatch journaux
- Supprimer le fichier original dans le compartiment source S3
- Supprimer le compartiment source S3
- Supprimer le rôle IAM de votre fonction Lambda
- Supprimer la politique gérée par le client de votre utilisateur IAM
- Supprimer l'utilisateur IAM

Supprimer le point d'accès Object Lambda

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Sur la page Points d'accès Lambda d'objet, choisissez le bouton d'option situé à gauche du point d'accès S3 Object Lambda que vous avez créé à l'étape 5 (par exemple, **`tutorial-pii-object-lambda-accesspoint`**).
4. Choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer votre point d'accès Object Lambda en saisissant son nom dans le champ de texte qui s'affiche, puis choisissez Supprimer.

Supprimer le point d'accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation, choisissez Points d'accès.
3. Accédez au point d'accès que vous avez créé à l'étape 3(par exemple, **`tutorial-pii-access-point`**) et choisissez le bouton d'option en regard du nom du point d'accès.
4. Choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer votre point d'accès en saisissant son nom dans le champ de texte qui s'affiche, puis choisissez Supprimer.

Supprimer la fonction Lambda

1. Dans la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/, choisissez Functions dans le volet de navigation de gauche.
2. Choisissez la fonction que vous avez créée à l'étape 4 (par exemple, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Choisissez Actions, puis choisissez Supprimer.
4. Dans la boîte de dialogue Supprimer une fonction, choisissez Supprimer.

Supprimer le groupe de CloudWatch journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Groupes de journaux.
3. Recherchez le groupe de journaux dont le nom se termine par la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
4. Choisissez Actions, puis choisissez Supprimer le groupe de journaux.
5. Dans la boîte de dialogue Supprimer le ou les groupes de journaux, choisissez Supprimer.

Supprimer le fichier original dans le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Nom du compartiment, choisissez le nom du compartiment vers lequel vous avez chargé le fichier original à l'étape 2 (par exemple, **`tutorial-bucket`**).
4. Cochez la case située à gauche du nom de l'objet que vous souhaitez supprimer (par exemple, `tutorial.txt`).
5. Choisissez Supprimer.
6. Dans la page Supprimer des objets, dans la section Supprimer définitivement les objets ?, confirmez que vous souhaitez supprimer cet objet en saisissant **`permanently delete`** dans la zone de texte.
7. Choisissez Supprimer les objets.

Supprimer le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le bouton d'option en regard du nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**).
4. Choisissez Supprimer.
5. Dans la page Supprimer un compartiment, confirmez que vous souhaitez supprimer le compartiment en saisissant le nom du compartiment dans le champ de texte, puis choisissez Supprimer un compartiment.

Supprimer le rôle IAM de votre fonction Lambda

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Rôles, puis cochez la case en regard du rôle à supprimer. Le nom du rôle commence par le nom de la fonction Lambda que vous avez déployée à l'étape 4 (par exemple, **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**).
3. Choisissez Supprimer.
4. Dans la boîte de dialogue Supprimer, saisissez le nom du rôle dans le champ de saisie de texte pour confirmer la suppression. Ensuite, choisissez Supprimer.

Supprimer la politique gérée par le client de votre utilisateur IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Dans la page Politiques, saisissez le nom de la politique gérée par le client que vous avez créée dans les conditions préalables (par exemple, **`tutorial-serverless-application-repository`**) dans la zone de recherche pour filtrer la liste des politiques. Sélectionnez le bouton d'option en regard du nom de la politique à supprimer.
4. Choisissez Actions, puis choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer cette politique en saisissant son nom dans le champ de texte qui s'affiche, puis choisissez Supprimer.

Supprimer l'utilisateur IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Utilisateurs, puis cochez la case en regard du nom de l'utilisateur à supprimer.
3. En haut de la page, choisissez Supprimer.
4. Dans la boîte de dialogue Supprimer ***`un nom d'utilisateur`*** ?, saisissez le nom de l'utilisateur dans le champ de saisie de texte afin de confirmer la suppression de l'utilisateur. Choisissez Supprimer.

Étapes suivantes

Après avoir terminé ce didacticiel, vous pouvez explorer plus en détail les cas d'utilisation connexes suivants :

- Vous pouvez créer plusieurs points d'accès S3 Lambda Object et les activer avec des fonctions Lambda préconstruites qui sont configurées différemment pour expurger des types spécifiques de DPI en fonction des besoins métier des accesseurs aux données.

  Chaque type d'utilisateur assume un rôle IAM et n'a accès qu'à un seul point d'accès S3 Object Lambda (géré via des politiques IAM). Ensuite, vous attachez chaque fonction Lambda `ComprehendPiiRedactionS3ObjectLambda` configurée pour un cas d'utilisation d'expurgation

différent à un autre point d'accès S3 Object Lambda. Pour chaque point d'accès S3 Object Lambda, vous pouvez disposer d'un point d'accès S3 prenant en charge la lecture des données à partir d'un compartiment S3 qui stocke le jeu de données partagé.

Pour en savoir plus sur la façon de créer une politique de compartiment S3 qui permet aux utilisateurs de lire à partir du compartiment uniquement via des points d'accès S3, consultez Configuration des stratégies IAM pour l'utilisation des points d'accès.

Pour en savoir plus sur la façon d'octroyer à un utilisateur l'autorisation d'accéder à la fonction Lambda, au point d'accès S3 et au point d'accès S3 Object Lambda, consultez Configuration des politiques IAM pour les points d'accès Object Lambda.

- Vous pouvez créer votre propre fonction Lambda et utiliser S3 Object Lambda avec votre fonction Lambda personnalisée pour répondre à vos besoins spécifiques en matière de données.

  Par exemple, pour explorer diverses valeurs de données, vous pouvez utiliser S3 Object Lambda et votre propre fonction Lambda qui utilise des fonctions Amazon Comprehend, telles que la reconnaissance des entités, la reconnaissance des mots clés, l'analyse du ressenti et la classification des documents, pour traiter les données. Vous pouvez également utiliser S3 Object Lambda avec Amazon Comprehend Medical, un service NLP admissible à HIPAA, pour analyser et extraire des données d'une manière contextuelle.

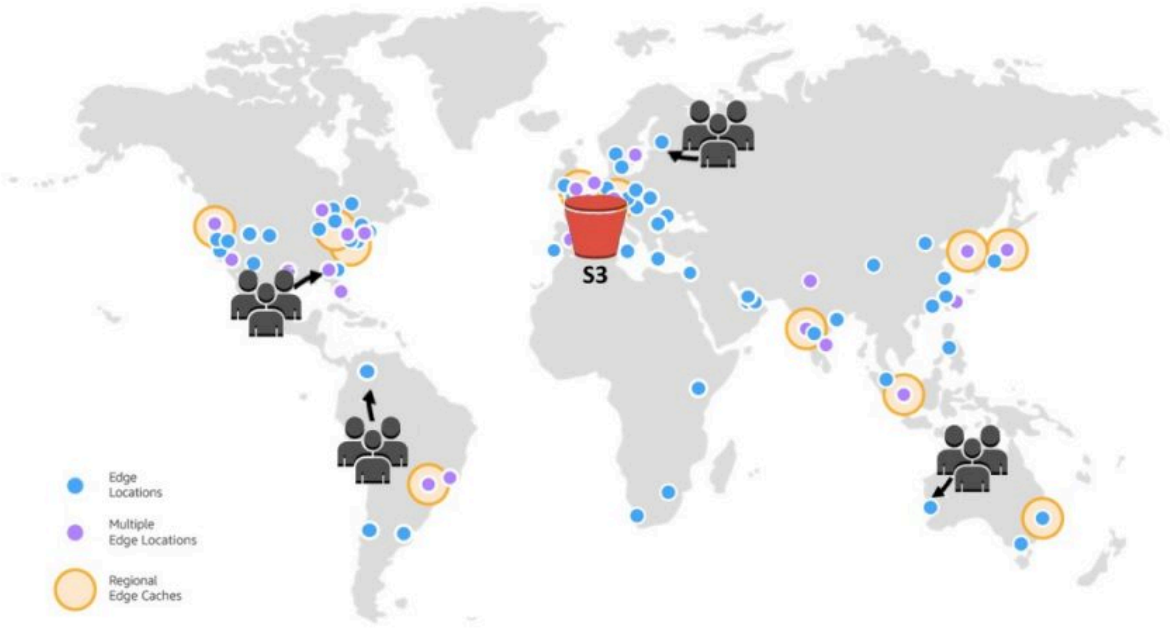
  Pour en savoir plus sur la façon de transformer des données avec S3 Object Lambda et votre propre fonction Lambda, consultez Didacticiel : Transformation de données pour votre application avec S3 Object Lambda.

Tutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53

Vous pouvez utiliser Amazon S3 avec Amazon CloudFront pour héberger des vidéos à visionner à la demande de manière sécurisée et évolutive. Les vidéos en streaming à la demande (VOD) signifient votre contenu vidéo est stocké sur un serveur et que les utilisateurs peuvent les regarder à tout moment.

CloudFront est un service de réseau de diffusion de contenu (CDN) rapide, hautement sécurisé et programmable. CloudFront peut diffuser votre contenu en toute sécurité via HTTPS à partir de tous les sites CloudFront périphériques du monde entier. Pour plus d'informations CloudFront, consultez Qu'est-ce qu'Amazon CloudFront ? dans le manuel Amazon CloudFront Developer Guide.

CloudFront la mise en cache réduit le nombre de demandes auxquelles votre serveur d'origine doit répondre directement. Lorsqu'un spectateur (utilisateur final) demande une vidéo que vous diffusez CloudFront, la demande est acheminée vers un emplacement périphérique proche de l'endroit où se trouve le spectateur. CloudFront diffuse la vidéo depuis son cache, en la récupérant du compartiment S3 uniquement si elle n'est pas déjà mise en cache. Cette fonction de gestion de la mise en cache accélère la livraison de vos vidéos aux utilisateurs du monde entier avec une faible latence, un débit élevé et des vitesses de transfert élevées. Pour plus d'informations sur la gestion de CloudFront la mise en cache, consultez Optimisation de la mise en cache et de la disponibilité dans le manuel Amazon CloudFront Developer Guide.



Objectif

Dans ce didacticiel, vous allez configurer un compartiment S3 pour héberger le streaming vidéo à la demande à l'aide CloudFront d'Amazon Route 53 pour le système de noms de domaine (DNS) et de la gestion personnalisée des domaines.

Rubriques

- Conditions préalables : enregistrez et configurez un domaine personnalisé avec Route 53
- Étape 1 : Créer un compartiment S3
- Étape 2 : Charger une vidéo dans le compartiment S3

Conditions préalables : enregistrez et configurez un domaine personnalisé avec Route 53

Avant de commencer ce didacticiel, vous devez enregistrer et configurer un domaine personnalisé (par exemple,**`example.com`**) avec Route 53 afin de pouvoir configurer votre CloudFront distribution pour utiliser un nom de domaine personnalisé ultérieurement.

Sans nom de domaine personnalisé, votre vidéo S3 est accessible CloudFront au public et hébergée via une URL similaire à la suivante :

```
https://CloudFront distribution domain name/Path to an S3 video
```

Par exemple, **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Une fois que vous avez configuré votre CloudFront distribution pour utiliser un nom de domaine personnalisé configuré avec Route 53, votre vidéo S3 est accessible CloudFront au public et hébergée via une URL similaire à la suivante :

```
https://CloudFront distribution alternate domain name/Path to an S3 video
```

Par exemple, **`https://www.example.com/sample.mp4`**. Un nom de domaine personnalisé est plus simple et plus intuitif pour les utilisateurs.

Pour enregistrer un domaine personnalisé, veuillez consulter Enregistrement d'un nouveau domaine avec Route 53 dans le Guide du développeur Amazon Route 53.

Si vous enregistrez un nom de domaine avec Route 53, Route 53 créera pour vous la zone hébergée que vous utiliserez ultérieurement dans ce didacticiel. Cette zone hébergée est l'endroit où vous stockez des informations sur la manière d'acheminer le trafic de votre domaine, par exemple vers une instance Amazon EC2 ou une CloudFront distribution.

Des frais sont associés à l'enregistrement de domaine, à votre zone hébergée et aux requêtes DNS reçues par votre domaine. Pour en savoir plus, consultez Tarification Amazon Route 53.

> **Note**
>
> Lorsque vous enregistrez un domaine, cela coûte immédiatement de l'argent et c'est irréversible. Vous pouvez choisir de ne pas renouveler automatiquement le domaine, mais vous payez à l'avance et le possédez pour l'année. Pour plus d'informations, consultez Enregistrement d'un nouveau domaine dans le Guide du développeur Amazon Route 53.

Étape 1 : Créer un compartiment S3

Créez un compartiment destiné à stocker la vidéo d'origine que vous prévoyez de diffuser.

Pour créer un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
4. Pour Bucket Name (Nom du compartiment), indiquez le nom de votre compartiment, (par exemple, **`tutorial-bucket`**).

   Pour en savoir plus sur les règles d'attribution de noms des compartiments Amazon S3, consultez Règles de dénomination de compartiment.
5. Pour Région, choisissez l' Région AWS endroit où vous souhaitez que le compartiment réside.

   Si possible, choisissez la région qui est la plus proche de la majorité de vos utilisateurs. Pour en savoir plus sur les régions des compartiments, consultez Présentation des compartiments.

6. Pour Paramètres de blocage de l'accès public à ce compartiment, conservez les paramètres par défaut (Bloquer tout accès public est activé).

   Même si l'option Bloquer tout accès public est activée, les spectateurs peuvent toujours accéder à la vidéo mise en ligne par le biais de cette option CloudFront. Cette fonctionnalité est un avantage majeur de l'utilisation CloudFront pour héberger une vidéo stockée dans S3.

   Nous vous recommandons de laisser tous les paramètres activés, sauf si vous devez en désactiver un ou plusieurs pour votre cas d'utilisation. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.
7. Pour les paramètres restants, conservez les paramètres par défaut.

   (Facultatif) Si vous souhaitez configurer des paramètres de compartiment supplémentaires pour votre cas d'utilisation particulier, consultez Créer un compartiment.
8. Choisissez Créer un compartiment.

Étape 2 : Charger une vidéo dans le compartiment S3

La procédure suivante explique la manière de télécharger un fichier vidéo dans un compartiment S3 à l'aide de la console. Lorsque vous téléchargez de nombreux fichiers vidéo volumineux sur S3, vous pouvez également utiliser Amazon S3 Transfer Acceleration pour configurer des transferts de fichiers rapides et sécurisés. Transfer Acceleration peut accélérer le chargement de vidéos vers votre compartiment S3 pour le transfert à longue distance de vidéos plus volumineuses. Pour plus d'informations, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Charger un fichier dans le compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**) pour y charger votre fichier.
4. Sous l'onglet Objets de votre compartiment, choisissez Charger.
5. Dans la page Charger, sous Fichiers et dossiers, choisissez Ajouter des fichiers.
6. Choisissez un fichier à charger, puis choisissez Ouvrir.

Par exemple, vous pouvez charger un fichier vidéo nommé `sample.mp4`.

7. Choisissez Charger.

Étape 3 : créer une identité d'accès à l' CloudFront origine

Pour restreindre l'accès direct à la vidéo depuis votre compartiment S3, créez un CloudFront utilisateur spécial appelé identité d'accès d'origine (OAI). Vous allez associer l'OAI à votre distribution plus tard au cours de ce didacticiel. En utilisant un OAI, vous vous assurez que les spectateurs ne peuvent pas contourner CloudFront et accéder à la vidéo directement depuis le compartiment S3. Seul l' CloudFront OAI peut accéder au fichier dans le compartiment S3. Pour plus d'informations, consultez Restreindre l'accès au contenu Amazon S3 à l'aide d'un OAI dans le manuel Amazon CloudFront Developer Guide.

Pour créer un CloudFront OAI

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans la section Sécurité du panneau de navigation de gauche, choisissez Accès à l'origine.
3. Sous l'onglet Identités, choisissez Créer une identité d'accès à l'origine.
4. Saisissez un nom (par exemple, **`S3-OAI`**) pour la nouvelle identité d'accès à l'origine.
5. Choisissez Créer.

Étape 4 : Création d'une CloudFront distribution

Pour l'utiliser CloudFront pour diffuser et distribuer la vidéo dans votre compartiment S3, vous devez créer une CloudFront distribution.

Sous-étapes

- Création d'une CloudFront distribution
- Examiner la politique de compartiment

Création d'une CloudFront distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.

2. Dans le panneau de navigation de gauche, choisissez Distributions.
3. Choisissez Create distribution (Créer une distribution).
4. Dans la section Origin (Origine), pour Origin domain (Domaine d'origine), choisissez le nom de domaine de votre origine S3, qui commence par le nom du compartiment S3 que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**).
5. Pour Accès d'origine, choisissez Identités d'accès existantes.
6. Sous Origin access identity (Identité d'accès à l'origine), choisissez l'identité d'accès à l'origine existante que vous avez créée à l'étape 3 (par exemple, **`S3-OAI`**).
7. Sous Politique de compartiment, choisissez Oui, mettre à jour la politique de compartiment.
8. Dans la section Default cache behavior (Comportement du cache par défaut), sous Viewer protocol policy (Politique du protocole de l'utilisateur), choisissez Redirect HTTP to HTTPS (Rediriger HTTP vers HTTPS).

   Lorsque vous choisissez cette fonction, les requêtes HTTP sont automatiquement redirigées vers HTTPS pour sécuriser votre site web et protéger les données de vos utilisateurs.
9. Pour les autres paramètres de la section Default cache behaviors (Comportement du cache par défaut), conservez les valeurs par défaut.

   (Facultatif) Vous pouvez contrôler la durée pendant laquelle votre fichier reste dans CloudFront le cache avant CloudFront de transmettre une autre demande à votre source. Réduire la durée vous permet de servir des contenus dynamiques. Augmenter la durée signifie que vos utilisateurs obtiennent de meilleures performances parce que vos fichiers sont plus susceptibles d'être servis directement à partir du cache périphérique. Une durée plus longue réduit également la charge sur votre origine. Pour plus d'informations, consultez Gérer la durée pendant laquelle le contenu reste dans le cache (expiration) dans le manuel Amazon CloudFront Developer Guide.
10. Pour les autres sections, conservez les paramètres restants définis sur les valeurs par défaut.

    Pour plus d'informations sur les différentes options de configuration, consultez la section Valeurs que vous spécifiez lorsque vous créez ou mettez à jour une distribution dans le manuel Amazon CloudFront Developer Guide.
11. En bas de la page, choisissez Créer une distribution.
12. Dans l'onglet Général de votre CloudFront distribution, sous Détails, la valeur de la colonne Dernière modification de votre distribution passe de Déploiement à l'horodatage de la dernière modification de la distribution. Ce processus prend généralement quelques minutes.

Examiner la politique de compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste des compartiments, choisissez le nom du compartiment que vous avez utilisé précédemment comme origine de votre CloudFront distribution (par exemple,**`tutorial-bucket`**).
4. Choisissez l'onglet Permissions (Autorisations).
5. Dans la boîte de dialogue Bucket policy (Stratégie de compartiment), confirmez que vous voyez une instruction similaire à ce qui suit s'affiche dans le texte de la stratégie de compartiment :

```
{
    "Version": "2008-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::tutorial-bucket/*"
        }
    ]
}
```

Il s'agit de la déclaration que votre CloudFront distribution a ajoutée à votre politique de compartiment lorsque vous avez choisi Oui, mettez à jour la politique de compartiment plus tôt.

Cette mise à jour de la politique de compartiment indique que vous avez correctement configuré la CloudFront distribution pour restreindre l'accès au compartiment S3. En raison de cette restriction, les objets du compartiment ne sont accessibles que par le biais de votre CloudFront distribution.

Étape 5 : Accédez à la vidéo par le biais de la CloudFront distribution

CloudFront Vous pouvez désormais diffuser la vidéo stockée dans votre compartiment S3. Pour accéder à votre vidéo CloudFront, vous devez associer le nom de votre domaine de CloudFront distribution au chemin d'accès à la vidéo dans le compartiment S3.

Pour créer une URL vers la vidéo S3 à l'aide du nom CloudFront de domaine de distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans le panneau de navigation de gauche, choisissez Distributions.
3. Pour obtenir le nom de domaine de distribution, procédez comme suit :
   a. Dans la colonne Origins, recherchez la CloudFront distribution correcte en recherchant son nom d'origine, qui commence par le compartiment S3 que vous avez créé à l'étape 1 (par exemple,**`tutorial-bucket`**).
   b. Après avoir trouvé la distribution dans la liste, élargissez la colonne Nom de domaine pour copier la valeur du nom de domaine de votre CloudFront distribution.
4. Dans un nouvel onglet du navigateur, collez le nom de domaine de distribution que vous avez copiés.
5. Revenez à l'onglet précédent du navigateur et ouvrez la console S3 à l'adresse https://console.aws.amazon.com/s3/.
6. Dans le panneau de navigation de gauche, choisissez Compartiments.
7. Dans la liste Compartiments, choisissez le nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**).
8. Dans la liste Objects (Objets), choisissez le nom de la vidéo que vous avez téléchargée à l'étape 2 (par exemple, `sample.mp4`).
9. Sur la page détaillée de l'objet, dans la section Object overview (Présentation de l'objet), copiez la valeur de la clé. Cette valeur correspond au chemin d'accès à l'objet vidéo chargé dans le compartiment S3.
10. Revenez à l'onglet du navigateur dans lequel vous avez précédemment collé le nom de domaine de distribution, saisissez une barre oblique (**`/`**) après le nom de domaine de distribution, puis collez le chemin d'accès à la vidéo que vous avez copiée précédemment (par exemple, `sample.mp4`).

Votre vidéo S3 est désormais accessible CloudFront au public et hébergée via une URL similaire à la suivante :

```
https://CloudFront distribution domain name/Path to the S3 video
```

Remplacez le *nom de domaine de CloudFront distribution* et *le chemin d'accès à la vidéo S3* par les valeurs appropriées. Voici un exemple d'URL : **https://d111111abcdef8.cloudfront.net/sample.mp4**

Étape 6 : Configurez votre CloudFront distribution pour utiliser votre nom de domaine personnalisé

Pour utiliser votre propre nom de domaine au lieu du nom de CloudFront domaine indiqué dans l'URL pour accéder à la vidéo S3, ajoutez un autre nom de domaine à votre CloudFront distribution.

Sous-étapes

- Demander un certificat SSL
- Ajoutez le nom de domaine alternatif à votre CloudFront distribution
- Créez un enregistrement DNS pour acheminer le trafic de votre nom de domaine alternatif vers le nom de domaine de votre CloudFront distribution
- Vérifiez si IPv6 est activé pour votre distribution et créez un autre registre DNS si nécessaire

Demander un certificat SSL

Pour permettre à vos spectateurs d'utiliser le protocole HTTPS et votre nom de domaine personnalisé dans l'URL de votre diffusion vidéo, utilisez AWS Certificate Manager (ACM) pour demander un certificat SSL (Secure Sockets Layer). Le certificat SSL établit une connexion réseau chiffrée au site web.

1. Connectez-vous à la console ACM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/acm/.
2. Si la page d'introduction s'affiche, sous Allocation de certificats, choisissez Mise en route.
3. Dans la page Request a certificate (Demander un certificat), choisissez Request a public certificate (Demander un certificat public) et Request a certificate (Demander un certificat).

4. Dans la page Add domain names (Ajouter des noms de domaine), saisissez le nom de domaine complet (FQDN) du site que vous souhaitez sécuriser à l'aide d'un certificat SSL/TLS. Vous pouvez utiliser un astérisque (*) pour créer un certificat générique qui protègera plusieurs noms de site du même domaine. Plus précisément, dans ce didacticiel, saisissez ***** et le nom de domaine personnalisé que vous avez configuré dans Prerequisites (Conditions préalables). Pour cet exemple, saisissez **`*.example.com`**, puis choisissez Next (Suivant).

   Pour en savoir plus, consultez Demander un certificat public ACM (console) dans le guide de l'utilisateur AWS Certificate Manager .

5. Sur la page Sélectionner une méthode de validation, choisissez validation du DNS. Ensuite, choisissez Suivant.

   Si vous êtes en mesure de modifier la configuration DNS, nous vous recommandons d'utiliser la validation de domaine DNS plutôt que la validation par e-mail. La validation du DNS présente plusieurs avantages par rapport à la validation par e-mail. Pour plus d'informations, consultez Option 1 : validation DNS dans le Guide de l'utilisateur AWS Certificate Manager .

6. (Facultatif) Étiquetez votre certificat à l'aide de métadonnées sur la page Ajouter des balises.

7. Choisissez Examiner.

8. Sur la page Vérification, vérifiez que les informations sous Domain name (Nom de domaine) et Validation method (Méthode de validation) sont correctes. Ensuite, choisissez Confirmer et demander.

   La page Validation indique que votre demande est en cours de traitement et que les domaines de certificat sont en cours de validation. Le certificat en attente de validation est dans l'état En attente de validation.

9. Dans la page Validation, appuyez sur la flèche vers le bas à gauche de votre nom de domaine personnalisé, puis choisissez Create record in Route 53 (Créer un registre dans Route 53) pour valider la propriété du domaine via le DNS.

   Cela ajoute un enregistrement CNAME fourni par AWS Certificate Manager à votre configuration DNS.

10. Dans la boîte de dialogue Créer un enregistrement dans Route 53, choisissez Créer.

    La page Validation devrait afficher une notification d'état Success (Réussite) en bas.

11. Choisissez Continuer pour afficher la page de la liste des certificats.

L'état de votre nouveau certificat passera de Pending validation (En attente de validation) à Issued (Émis) dans les 30 minutes.

Ajoutez le nom de domaine alternatif à votre CloudFront distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans le panneau de navigation de gauche, choisissez Distributions.
3. Choisissez l'ID de la distribution que vous avez créée à l'étape 4.
4. Sur la page General (Général), accédez à l'onglet Settings (Paramètres) et choisissez Edit (Modifier).
5. Sur la page Modifier les paramètres, pour Nom de domaine alternatif (CNAME) - facultatif, choisissez Ajouter un élément pour ajouter les noms de domaine personnalisés que vous souhaitez utiliser dans l'URL de la vidéo S3 diffusée par cette CloudFront distribution.

   Dans ce didacticiel, par exemple, si vous souhaitez acheminer le trafic d'un sous-domaine, tel que `www.example.com`, saisissez le nom de sous-domaine (`www`) avec le nom de domaine (`example.com`). Plus précisément, saisissez **`www.example.com`**.

   > **Note**
   >
   > Le nom de domaine alternatif (CNAME) que vous ajoutez doit être couvert par le certificat SSL que vous avez précédemment attaché à votre CloudFront distribution.

6. Pour Custom SSL certificate - optional (Certificat SSL personnalisé - facultatif), choisissez le certificat SSL que vous avez demandé précédemment (par exemple, **`*.example.com`**).

   > **Note**
   >
   > Si le certificat SSL ne s'affiche pas immédiatement après l'avoir demandé, patientez 30 minutes, puis actualisez la liste jusqu'à ce que le certificat SSL soit disponible afin que vous puissiez le sélectionner.

7. Conservez les paramètres restants définis sur les valeurs par défaut. Choisissez Enregistrer les modifications.

8. Dans l'onglet General (Généra)l de la distribution, attendez que la valeur de Last modified (Dernière modification) passe de Deploying (Déploiement) à l'horodatage du moment où la distribution a été modifiée pour la dernière fois.

Créez un enregistrement DNS pour acheminer le trafic de votre nom de domaine alternatif vers le nom de domaine de votre CloudFront distribution

1. Connectez-vous à la console Route 53 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/route53/.
2. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).
3. Dans la page Zones hébergées, choisissez le nom de la zone hébergée que Route 53 a créée pour vous dans les conditions préalables (par exemple, **`example.com`**).
4. Choisissez Create record (Créer un registre), puis utilisez la méthode Quick create record (Création rapide de registre).
5. Pour Nom de l'enregistrement, conservez la même valeur que le nom de domaine alternatif de la CloudFront distribution que vous avez ajouté précédemment.

   Dans ce didacticiel, pour acheminer le trafic vers un sous-domaine, tel que `www.example.com`, saisissez le nom de sous-domaine sans le nom de domaine. Par exemple, saisissez uniquement **`www`** dans le champ de texte avant votre nom de domaine personnalisé.
6. Pour Type d'enregistrement, choisissez A - Route le trafic vers une adresse IPv4 et certaines AWS ressources.
7. Pour Value (Valeur), choisissez Alias pour activer la ressource Alias.
8. Sous Acheminer le trafic vers, choisissez Alias vers la CloudFront distribution dans la liste déroulante.
9. Dans la zone de recherche intitulée Choisir une distribution, choisissez le nom de domaine de la CloudFront distribution que vous avez créée à l'étape 4.

   Pour trouver le nom de domaine de votre CloudFront distribution, procédez comme suit :

   a. Dans un nouvel onglet du navigateur, connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v3/home.

   b. Dans le panneau de navigation de gauche, choisissez Distributions.

c. Dans la colonne Origins, recherchez la CloudFront distribution correcte en recherchant son nom d'origine, qui commence par le compartiment S3 que vous avez créé à l'étape 1 (par exemple,**`tutorial-bucket`**).

d. Après avoir trouvé la distribution dans la liste, élargissez la colonne Nom de domaine pour voir la valeur du nom de domaine de votre CloudFront distribution.

10. Sur la page Create record (Création de registre) de la console Route 53, conservez les valeurs par défaut pour les paramètres restants.

11. Choisissez Create records (Créer des registres).

Vérifiez si IPv6 est activé pour votre distribution et créez un autre registre DNS si nécessaire

Si IPv6 est activé pour votre distribution, vous devez créer un autre registre DNS.

1. Pour vérifier si IPv6 est activé pour votre distribution, procédez comme suit :

   a. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.

   b. Dans le panneau de navigation de gauche, choisissez Distributions.

   c. Choisissez l'ID de la CloudFront distribution que vous avez créée à l'étape 4.

   d. Sur l'onglet General (Général), sous Settings (Paramètres), vérifiez si IPv6 est défini sur Enabled (Activé).

   Si IPv6 est activé pour votre distribution, vous devez créer un autre registre DNS.

2. Si IPv6 est activé pour votre distribution, procédez comme suit pour créer un registre DNS :

   a. Connectez-vous à la console Route 53 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/route53/.

   b. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).

   c. Dans la page Zones hébergées, choisissez le nom de la zone hébergée que Route 53 a créée pour vous dans les conditions préalables (par exemple, **`example.com`**).

   d. Choisissez Create record (Créer un registre), puis utilisez la méthode Quick create record (Création rapide de registre).

   e. Pour Record name (Nom du registre), dans le champ de texte précédant votre nom de domaine personnalisé, saisissez la même valeur que celle que vous avez saisie lors de la

création de registre DNS IPv4 ci-dessus. Par exemple, dans ce didacticiel, pour acheminer le trafic pour le sous-domaine `www.example.com`, saisissez uniquement **`www`**.

f. Pour Type de registre, choisissez AAAA – Achemine le trafic vers une adresse IPv6 et certaines ressources AWS .

g. Pour Value (Valeur), choisissez Alias pour activer la ressource Alias.

h. Sous Acheminer le trafic vers, choisissez Alias vers la CloudFront distribution dans la liste déroulante.

i. Dans la zone de recherche intitulée Choisir une distribution, choisissez le nom de domaine de la CloudFront distribution que vous avez créée à l'étape 4.

j. Pour les paramètres restants, conservez les paramètres par défaut.

k. Choisissez Create records (Créer des registres).

Étape 7 : Accédez à la vidéo S3 via la CloudFront distribution avec le nom de domaine personnalisé

Pour accéder à la vidéo S3 à l'aide de l'URL personnalisée, vous devez associer votre autre nom de domaine au chemin d'accès à la vidéo dans le compartiment S3.

Pour créer une URL personnalisée permettant d'accéder à la vidéo S3 via la CloudFront distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans le panneau de navigation de gauche, choisissez Distributions.
3. Pour obtenir le nom de domaine alternatif de votre CloudFront distribution, procédez comme suit :

   a. Dans la colonne Origins, recherchez la CloudFront distribution correcte en recherchant son nom d'origine, qui commence par le nom du compartiment S3 que vous avez créé à l'étape 1 (par exemple,**`tutorial-bucket`**).

   b. Après avoir trouvé la distribution dans la liste, élargissez la colonne Noms de domaine alternatifs pour copier la valeur du nom de domaine alternatif de votre CloudFront distribution.
4. Dans un nouvel onglet du navigateur, collez le nom de domaine alternatif de la CloudFront distribution.

5. Revenez à l'onglet précédent du navigateur et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
6. Recherchez le chemin d'accès à votre vidéo S3, comme expliqué à l'étape 5.
7. Retournez à l'onglet du navigateur où vous avez précédemment collé le nom de domaine alternatif, saisissez une barre oblique (**/**) et collez le chemin d'accès à votre vidéo S3 (par exemple, `sample.mp4`).

   Votre vidéo S3 est désormais accessible au public et hébergée via CloudFront une URL personnalisée qui ressemble à ce qui suit :

   ```
   https://CloudFront distribution alternate domain name/Path to the S3 video
   ```

   Remplacez le *nom de domaine alternatif de CloudFront distribution* et *le chemin d'accès à la vidéo S3* par les valeurs appropriées. Voici un exemple d'URL : **https://www.example.com/sample.mp4**

(Facultatif) Étape 8 : Afficher les données relatives aux demandes reçues par votre CloudFront distribution

Pour consulter les données relatives aux demandes reçues par votre CloudFront distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans le panneau de navigation gauche, sous Reports & analytics (Rapports et analytique), choisissez les rapports dans la console, allant de Cache statistics (Statistiques du cache), Popular Objects (Objets populaires), Top Referrers (Principaux référents), Usage (Utilisation) et Viewers (Utilisateurs).

   Vous pouvez filtrer le tableau de bord de chaque rapport. Pour plus d'informations, consultez la section CloudFront Rapports de la console dans le manuel Amazon CloudFront Developer Guide.
3. Pour filtrer les données, choisissez l'ID de la CloudFront distribution que vous avez créée à l'étape 4.

Étape 9 : Nettoyer

Si vous avez hébergé une vidéo S3 en streaming en utilisant CloudFront Route 53 uniquement à titre d'exercice d'apprentissage, supprimez les AWS ressources que vous avez allouées afin de ne plus accumuler de frais.

> **Note**
>
> Lorsque vous enregistrez un domaine, cela coûte immédiatement de l'argent et c'est irréversible. Vous pouvez choisir de ne pas renouveler automatiquement le domaine, mais vous payez à l'avance et le possédez pour l'année. Pour plus d'informations, consultez Enregistrement d'un nouveau domaine dans le Guide du développeur Amazon Route 53.

Sous-étapes

- Supprimer la CloudFront distribution
- Suppression du registre DNS
- Supprimer la zone hébergée publique de votre domaine personnalisé
- Supprimer le nom de domaine personnalisé de Route 53
- Supprimer la vidéo d'origine dans le compartiment source S3
- Supprimer le compartiment source S3

Supprimer la CloudFront distribution

1. Connectez-vous à la CloudFront console AWS Management Console et ouvrez-la à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Dans le panneau de navigation de gauche, choisissez Distributions.
3. Dans la colonne Origins, recherchez la CloudFront distribution correcte en recherchant son nom d'origine, qui commence par le nom du compartiment S3 que vous avez créé à l'étape 1 (par exemple,**`tutorial-bucket`**).
4. Pour supprimer la CloudFront distribution, vous devez d'abord la désactiver.

   - Si la valeur de la colonne Status (État) est Enabled (Activé) et si la valeur Last modified (Dernière modification) est l'horodatage au moment de la dernière modification de la distribution, continuez à désactiver la distribution avant de la supprimer.

- Si la valeur de Status (État) est Enabled (Activé) et si la valeur Last modified (Dernière modification) est Deploying (Déploiement), attendez jusqu'à ce que la valeur de Status (État) passe à l'horodatage du moment où la distribution a été modifiée pour la dernière fois. Continuez ensuite pour désactiver la distribution avant de la supprimer.

5. Pour désactiver la CloudFront distribution, procédez comme suit :

   a. Dans la liste Distributions, cochez la case en regard de l'ID de la distribution que vous souhaitez supprimer.

   b. Pour désactiver la distribution, choisissez Disable (Désactiver), puis Delete (Oui, supprimer) pour confirmer.

      Si vous désactivez une distribution associée à un autre nom de domaine, elle CloudFront cesse d'accepter du trafic pour ce nom de domaine (par exemple`www.example.com`), même si une autre distribution possède un nom de domaine alternatif avec un caractère générique (`*`) correspondant au même domaine (tel que`*.example.com`).

   c. La valeur de la colonne État passe immédiatement sur Désactivé. Attendez que la valeur Last modified (Dernière modification) passe de Deploying (Déploiement) à l'horodatage du moment où la distribution a été modifiée pour la dernière fois.

      Comme cette modification CloudFront doit être étendue à tous les emplacements périphériques, cela peut prendre quelques minutes avant que la mise à jour ne soit terminée et que l'option Supprimer soit disponible pour vous permettre de supprimer la distribution.

6. Pour supprimer la distribution désactivée, procédez comme suit :

   a. Cochez la case en regard de l'ID de la distribution que vous souhaitez supprimer.

   b. Choisissez Delete (Supprimer), puis Delete (Oui, supprimer) pour confirmer.

Suppression du registre DNS

Si vous souhaitez supprimer la zone hébergée publique du domaine (y compris le registre DNS), veuillez consulter Supprimer la zone hébergée publique de votre domaine personnalisé dans le Guide du développeur Amazon Route 53. Si vous souhaitez seulement supprimer le registre DNS créé à l'étape 6, procédez comme suit :

1. Connectez-vous à la console Route 53 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/route53/.
2. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).

3. Dans la page Zones hébergées, choisissez le nom de la zone hébergée que Route 53 a créée pour vous dans les conditions préalables (par exemple, **`example.com`**).
4. Dans la liste des registres, cochez la case à côté des registres que vous voulez supprimer (les registres que vous avez créés à l'étape 6).

   > **Note**
   >
   > Vous ne pouvez pas supprimer les registres dont la valeur est NS ou SOA pour le champ Type.

5. Choisissez Supprimer un jeu de registres.
6. Pour confirmer la suppression, choisissez Supprimer.

   La propagation de vos registres modifiés sur les serveurs DNS Route 53 n'est pas immédiate. À l'heure actuelle, le seul moyen de vérifier que vos modifications se sont propagées consiste à utiliser l'action GetChange API. Les modifications se propagent généralement sur tous les serveurs de Route 53 en 60 secondes.

Supprimer la zone hébergée publique de votre domaine personnalisé

> **Warning**
>
> Si vous souhaitez conserver l'enregistrement de votre domaine, mais que vous souhaitez arrêter d'acheminer le trafic Internet vers votre site web ou votre application Web, nous vous recommandons de supprimer les registres dans la zone hébergée (comme expliqué dans la section précédente) au lieu de supprimer la zone hébergée.
> Si vous supprimez une zone hébergée, quelqu'un d'autre peut utiliser le domaine et acheminer le trafic vers ses propres ressources à l'aide de votre nom de domaine.
> En outre, si vous supprimez une zone hébergée, vous ne pourrez pas annuler cette suppression. Vous devrez créer une nouvelle zone hébergée et mettre à jour les serveurs de noms pour l'enregistrement de votre domaine, ce qui peut demander jusqu'à 48 heures pour prendre effet.
> Si vous ne voulez pas que le domaine soit disponible sur Internet, vous pouvez d'abord transférer votre service DNS vers un service DNS gratuit, puis supprimer la zone hébergée Route 53. Cela évitera les requêtes DNS futures d'être mal acheminées.

1. Si le domaine est enregistré auprès de Route 53, veuillez consulter Ajout ou modification de serveurs de noms et de registres de type Glue pour un domaine dans le Guide du développeur Amazon Route 53 pour savoir comment remplacer des serveurs de noms Route 53 par des serveurs de noms pour le nouveau service DNS.
2. Si le domaine est enregistré auprès d'un autre bureau d'enregistrement, utilisez la méthode fournie par le bureau d'enregistrement pour changer les serveurs de noms du domaine.

> **Note**
>
> Si vous supprimez une zone hébergée pour un sous-domaine (`www.example.com`), vous n'avez pas besoin de changer les serveurs de noms du domaine (`example.com`).

1. Connectez-vous à la console Route 53 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/route53/.
2. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).
3. Dans la page Zones hébergées, choisissez la ligne de la zone hébergée que vous souhaitez supprimer.
4. Sur l'onglet Registres de votre zone hébergée, vérifiez que la zone hébergée que vous souhaitez supprimer contient uniquement un NS et un registre SOA.

   Si elle contient d'autres registres, supprimez-les en premier lieu.

   Si vous avez créé des registres NS pour des sous-domaines dans la zone hébergée, supprimez également ces registres.
5. Sur l'onglet Signature DNSSEC de votre zone hébergée, désactivez la signature DNNSSEC si elle était activée. Pour plus d'informations, veuillez consulter Désactivation de la signature DNSSEC dans le Guide du développeur Amazon Route 53.
6. En haut de la page des détails de la zone hébergée, choisissez Delete zone (Supprimer une zone).
7. Saisissez **`delete`** pour confirmer la suppression, puis choisissez Supprimer.

Supprimer le nom de domaine personnalisé de Route 53

Pour les domaines de premier niveau (TLD), vous pouvez supprimer l'enregistrement si vous n'en avez plus besoin. Si vous supprimez un enregistrement de nom de domaine sur Route 53 avant l'expiration prévue de l'enregistrement, les frais d'enregistrement AWS ne sont pas remboursés. Pour en savoir plus, veuillez consulter Suppression d'un enregistrement de nom de domaine dans le Guide du développeur Amazon Route 53.

> **Important**
>
> Si vous souhaitez transférer le domaine entre Comptes AWS ou transférer le domaine vers un autre bureau d'enregistrement, ne supprimez pas le domaine et attendez-vous à le réenregistrer immédiatement. Consultez plutôt la documentation pertinente dans le Guide du développeur Amazon Route 53 :
>
> - Transférer un domaine vers un autre Compte AWS
> - Transfert d'un domaine depuis Amazon Route 53 vers un autre bureau d'enregistrement

Supprimer la vidéo d'origine dans le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Bucket name (Nom du compartiment), choisissez le nom du compartiment dans lequel vous avez téléchargé la vidéo à l'étape 2 (par exemple, **`tutorial-bucket`**).
4. Sur l'onglet Objects (Objets), cochez la case située en regard du nom de l'objet que vous souhaitez supprimer (par exemple, `sample.mp4`).
5. Sélectionnez Delete (Supprimer).
6. Sous Permanently delete objects? (Supprimer définitivement les objets ?), saisissez **`permanently delete`** pour confirmer que vous souhaitez supprimer cet objet.
7. Choisissez Supprimer les objets.

Supprimer le compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le bouton d'option en regard du nom du compartiment que vous avez créé à l'étape 1 (par exemple, **`tutorial-bucket`**).
4. Choisissez Supprimer.
5. Dans la page Supprimer le compartiment, confirmez que vous souhaitez supprimer le compartiment en saisissant le nom de ce dernier dans le champ de texte, puis choisissez Supprimer le compartiment.

Étapes suivantes

Après avoir terminé ce didacticiel, vous pouvez explorer plus en détail les cas d'utilisation connexes suivants :

- Transcodez les vidéos S3 dans les formats de streaming requis par un téléviseur ou un appareil connecté en particulier avant de les héberger dans une CloudFront distribution.

  Pour utiliser Amazon S3 Batch Operations AWS Lambda et AWS Elemental MediaConvert pour transcoder par lots une collection de vidéos vers différents formats de sortie, consultez. Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert
- Hébergez d'autres objets stockés dans S3, tels que des images, du son, des animations, des feuilles de style, du HTML JavaScript, des applications React, etc., à l'aide CloudFront de Route 53.

  Consultez, par exemple, Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53 et Accélérez votre site Web avec Amazon CloudFront.
- Utiliser Amazon S3 Transfer Acceleration pour configurer des transferts de fichiers rapides et sécurisés. Transfer Acceleration peut accélérer le chargement de vidéos vers votre compartiment S3 pour le transfert à longue distance de vidéos plus volumineuses. L'accélération des transferts améliore les performances de transfert en acheminant le trafic via les emplacements périphériques répartis CloudFront dans le monde entier et via AWS les réseaux principaux. Elle utilise également

des optimisations de protocole réseau. Pour plus d'informations, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

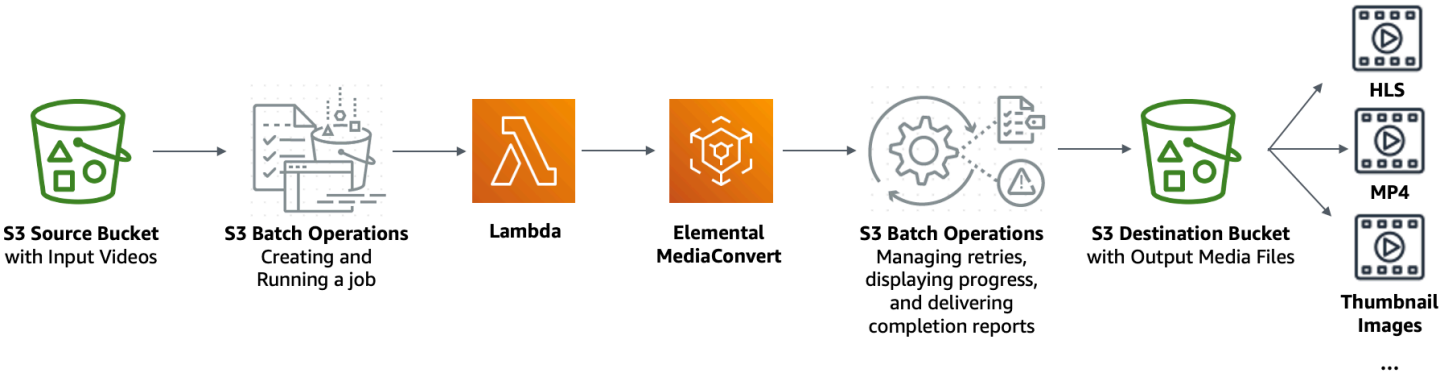
Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Les consommateurs de vidéos utilisent des appareils de toutes formes, tailles et millésimes pour profiter des contenus multimédia. Ce large éventail de services représente un défi pour les créateurs et distributeurs de contenus. Au lieu d'être dans un one-size-fits-all format, les vidéos doivent être converties afin qu'elles puissent couvrir une large gamme de tailles, de formats et de débits. Cette tâche de conversion est encore plus difficile lorsque vous avez un grand nombre de vidéos à convertir.

AWS vous propose une méthode pour créer une architecture distribuée évolutive qui effectue les opérations suivantes :

- Intégrer les vidéos d'entrée
- Traiter les vidéos en vue de leur lecture sur un large éventail d'appareils
- Stocker les fichiers multimédias transcodés
- Fournir les fichiers multimédias de sortie pour répondre à la demande

Si vous disposez de référentiels vidéo étendus stockés dans Amazon S3, vous pouvez transcoder ces vidéos à partir de leurs formats source en plusieurs types de fichiers dans la taille, la résolution et le format requis par un lecteur vidéo ou un appareil particulier. Plus précisément, S3 Batch Operations vous fournit une solution pour invoquer AWS Lambda des fonctions pour les vidéos d'entrée existantes dans un compartiment source S3. Ensuite, les fonctions Lambda appellent AWS Elemental MediaConvert pour effectuer des tâches de transcodage de vidéos à grande échelle. Les fichiers multimédias de sortie convertis sont stockés dans un compartiment de destination S3.



Objectif

Dans ce didacticiel, vous apprendrez à configurer S3 Batch Operations pour appeler une fonction Lambda pour le transcodage par lots de vidéos stockées dans un compartiment source S3. La fonction Lambda appelle MediaConvert pour transcoder les vidéos. Les sorties de chaque vidéo dans le compartiment source S3 se présentent comme suit :

- Un flux à débit adaptatif HTTP Live Streaming (HLS) pour lecture sur des appareils de diverses tailles et des largeurs de bande passante variables
- Un fichier vidéo MP4
- Images miniatures collectées à intervalles

Rubriques

Prérequis

Avant de commencer à suivre ce didacticiel, vous devez disposer d'un compartiment source Amazon S3 (par exemple **`tutorial-bucket-1`**) avec des vidéos à transcoder déjà stockées.

Vous pouvez donner un autre nom au compartiment si vous le souhaitez. Pour en savoir plus sur les règles d'attribution de noms de compartiment dans Amazon S3, consultez Règles de dénomination de compartiment.

Pour le compartiment source S3, conservez les paramètres associés aux Paramètres de blocage de l'accès public à ce compartiment définis pour les valeurs par défaut (Block all public access (Bloquer tous les accès publics) est activé). Pour plus d'informations, consultez Créer un compartiment.

Pour en savoir plus sur le téléchargement de vidéos vers le compartiment source S3, veuillez consulter Chargement d'objets. Lorsque vous téléchargez de nombreux fichiers vidéo volumineux sur S3, vous pouvez également utiliser Amazon S3 Transfer Acceleration pour configurer des transferts de fichiers rapides et sécurisés. Transfer Acceleration peut accélérer le chargement de vidéos vers votre compartiment S3 pour le transfert à longue distance de vidéos plus volumineuses. Pour plus d'informations, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Étape 1 : Créer un compartiment S3 pour les fichiers multimédias de sortie

Dans cette étape, vous créez un compartiment de destination S3 pour stocker les fichiers multimédias de sortie convertis. Vous créez également une configuration CORS (Cross Origin Resource Sharing) pour autoriser l'accès croisé aux fichiers multimédias transcodés stockés dans votre compartiment de destination S3.

Sous-étapes

- Créer un compartiment pour les fichiers multimédias de sortie
- Ajouter une configuration CORS à un compartiment de sortie S3

Créer un compartiment pour les fichiers multimédias de sortie

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.

3. Choisissez Créer un compartiment.
4. Pour Bucket Name (Nom du compartiment), indiquez le nom de votre compartiment, (par exemple, **`tutorial-bucket-2`**).
5. Pour Région, choisissez l' Région AWS endroit où vous souhaitez que le compartiment réside.
6. Pour garantir l'accès public à vos fichiers multimédias de sortie, dans Paramètres de blocage de l'accès public à ce compartiment, désactivez Bloquer tous les accès publics.

> **Warning**
>
> Avant de terminer cette étape, revoyez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tous les accès publics à vos compartiments.
> Si vous ne souhaitez pas effacer les paramètres de blocage de l'accès public, vous pouvez utiliser Amazon CloudFront pour transmettre les fichiers multimédia transcodés aux spectateurs (utilisateurs finaux). Pour plus d'informations, consultez Tutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53.

7. Cochez la case pour Je reconnais que les paramètres actuels pourraient rendre ce compartiment et les objets qu'il contient accessibles publiquement.
8. Conservez les paramètres restants définis sur les valeurs par défaut.
9. Choisissez Créer un compartiment.

Ajouter une configuration CORS à un compartiment de sortie S3

Une configuration JSON CORS définit un moyen pour les applications Web clientes (lecteurs vidéo dans ce contexte) chargées dans un domaine particulier de lire des fichiers multimédias de sortie transcodés dans un autre domaine.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.

3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous avez créé précédemment (par ex., **`tutorial-bucket-2`**).
4. Choisissez l'onglet Permissions (Autorisations).
5. Dans la section Partage des ressources cross-origin (CORS), choisissez Modifier.
6. Dans la zone de texte de configuration CORS, copiez-collez la configuration CORS ci-dessous.

   La configuration CORS doit être au format JSON. Dans cet exemple, l'attribut `AllowedOrigins` utilise le caractère générique (`*`) pour spécifier toutes les origines. Si vous connaissez votre origine précise, vous pouvez restreindre l'attribut `AllowedOrigins` à l'URL de votre lecteur spécifique. Pour plus d'informations sur la configuration de cet attribut et d'autres, veuillez consulter Configuration CORS.

```
[
    {
        "AllowedOrigins": [
            "*"
        ],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedHeaders": [
            "*"
        ],
        "ExposeHeaders": []

    }
]
```

7. Sélectionnez Enregistrer les modifications.

Étape 2 : créer un rôle IAM pour MediaConvert

AWS Elemental MediaConvert Pour pouvoir transcoder des vidéos d'entrée stockées dans votre compartiment S3, vous devez disposer d'un rôle de service AWS Identity and Access Management (IAM) qui octroie MediaConvert les autorisations de lecture et d'écriture de fichiers vidéo depuis et vers vos compartiments source et destination S3. Lorsque vous exécutez des tâches de transcodage, la MediaConvert console utilise ce rôle.

Pour créer un rôle IAM pour MediaConvert

1. Créez un rôle IAM avec un nom de rôle que vous choisissez (par exemple, **`tutorial-mediaconvert-role`**). Pour créer ce rôle, suivez les étapes décrites dans la section Créer votre MediaConvert rôle dans IAM (console) dans le guide de l'AWS Elemental MediaConvert utilisateur.
2. Après avoir créé le rôle IAM pour MediaConvert, dans la liste des rôles, choisissez le nom du rôle pour MediaConvert lequel vous avez créé (par exemple,**`tutorial-mediaconvert-role`**).
3. Dans la page Summary (Récapitulatif), copiez l'ARN du rôle (commençant par `arn:aws:iam::`), puis enregistrez l'ARN en vue d'une utilisation ultérieure.

   Pour en savoir plus sur les ARN, consultez la section Amazon Resource Names (ARN) dans les Références générales AWS .

Étape 3 : Créer un rôle IAM pour votre fonction Lambda

Pour transcoder des vidéos par lots avec MediaConvert S3 Batch Operations, vous utilisez une fonction Lambda pour connecter ces deux services afin de convertir des vidéos. Cette fonction Lambda doit avoir un rôle IAM qui accorde à la fonction Lambda des autorisations d'accès et MediaConvert des opérations de traitement par lots S3.

Sous-étapes

- Créer un rôle IAM pour votre fonction Lambda
- Intégrez une politique en ligne pour le rôle IAM de votre fonction Lambda

Créer un rôle IAM pour votre fonction Lambda

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Cliquez sur le type de rôle de service AWS , puis, sous Common use cases (Cas d'utilisation courants), choisissez Lambda.
4. Sélectionnez Next: Permissions (Étape suivante : autorisations).

5. Sur la page Attach Permissions policies (Attacher les stratégies d'autorisations), saisissez **`AWSLambdaBasicExecutionRole`** dans la case Filter policies (Politiques de filtrage). Pour associer la politique gérée AWSLambdaBasicExecutionRoleà ce rôle afin d'accorder des autorisations d'écriture à Amazon CloudWatch Logs, cochez la case à côté de AWSLambdaBasicExecutionRole.
6. Choisissez Suivant : Balises.
7. (Facultatif) Ajoutez des balises à la stratégie gérée.
8. Choisissez Next: Review (Suivant : Vérification).
9. Pour le Nom du rôle, saisissez **`tutorial-lambda-transcode-role`**.
10. Choisissez Créer un rôle.

Intégrez une politique en ligne pour le rôle IAM de votre fonction Lambda

Pour accorder des autorisations à la MediaConvert ressource nécessaire à l'exécution de la fonction Lambda, vous devez utiliser une politique intégrée.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Rôles.
3. Dans la liste Roles (Rôles), choisissez le nom du rôle IAM que vous avez créé précédemment pour votre fonction Lambda (par exemple, **`tutorial-lambda-transcode-role`**).
4. Sélectionnez l'onglet Autorisations.
5. Sélectionnez Ajouter une politique en ligne.
6. Choisissez l'onglet JSON, puis copiez et collez la stratégie JSON suivante.

   Dans la politique JSON, remplacez l'exemple de valeur `Resource` d'ARN de par le rôle ARN du rôle IAM MediaConvert que vous avez créé à l'étape 2 (par exemple,**`tutorial-mediaconvert-role`**).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
```

```
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Logging"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/tutorial-mediaconvert-role"
            ],
            "Effect": "Allow",
            "Sid": "PassRole"
        },
        {
            "Action": [
                "mediaconvert:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "MediaConvertService"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "S3Service"
        }
    ]
}
```

7. Choisissez Examiner une politique.
8. Pour Name (Nom), saisissez **`tutorial-lambda-policy`**.
9. Choisissez Créer une politique.

Lorsque vous aurez créé une politique en ligne, elle sera automatiquement intégrée au rôle IAM de votre fonction Lambda.

Étape 4 : Créer une fonction Lambda pour le transcodage vidéo

Dans cette section du didacticiel, vous allez créer une fonction Lambda à l'aide du SDK pour Python afin de l'intégrer à S3 Batch Operations et. MediaConvert Pour commencer à transcoder les vidéos déjà stockées dans votre compartiment source S3, vous exécutez une tâche S3 Batch Operations qui appelle directement la fonction Lambda pour chaque vidéo du compartiment source S3. Ensuite, la fonction Lambda soumet une tâche de transcodage pour chaque vidéo à. MediaConvert

Sous-étapes

- Écrire le code de la fonction Lambda et créer un package de déploiement
- Créer une fonction Lambda à l'aide d'un rôle d'exécution (console)
- Déployez votre fonction Lambda avec les archives de fichiers .zip, puis configurez la fonction Lambda (console)

Écrire le code de la fonction Lambda et créer un package de déploiement

1. Sur votre ordinateur local, créez un dossier nommé `batch-transcode`.
2. Dans le dossier `batch-transcode`, créez un fichier contenant les paramètres de la tâche JSON. Par exemple, vous pouvez utiliser les paramètres fournis dans cette section et nommer le fichier `job.json`.

   Un fichier `job.json` spécifie les éléments suivants :

   - Quels fichiers transcoder
   - Comment transcoder vos vidéos d'entrée
   - Quels fichiers multimédias de sortie vous souhaitez créer
   - Quel nom à attribuer aux fichiers transcodés
   - Où enregistrer les fichiers transcodés
   - Quelles fonctions avancées appliquer, et ainsi de suite

Dans ce didacticiel, nous utiliserons le fichier `job.json` suivant pour créer les sorties suivantes pour chaque vidéo dans le compartiment source S3 :

- Un flux à débit adaptatif HTTP Live Streaming (HLS) pour lecture sur plusieurs appareils de diverses tailles et des largeurs de bande passante variables
- Un fichier vidéo MP4
- Images miniatures collectées à intervalles

Ce fichier d'exemple `job.json` utilise le débit variable selon la qualité (QVCR) pour optimiser la qualité de la vidéo. La sortie HLS est conforme aux applications d'Apple (fichier audio séparé du fichier vidéo, durée de segment de 6 secondes et qualité vidéo optimisée via QVBR automatique).

Si vous ne souhaitez pas utiliser les exemples de paramètres fournis ici, vous pouvez générer une spécification `job.json` basée sur votre cas d'utilisation. Pour assurer la cohérence entre vos sorties, vérifiez que vos fichiers d'entrée ont des configurations vidéo et audio similaires. Créez des automatisations distinctes (paramètres `job.json` uniques) pour tous les fichiers d'entrée avec différentes configurations vidéo et audio. Pour en savoir plus, consultez Exemples de paramètres de tâche AWS Elemental MediaConvert en JSON dans le Guide de l'utilisateur AWS Elemental MediaConvert .

```
{
  "OutputGroups": [
    {
      "CustomName": "HLS",
      "Name": "Apple HLS",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
```

```
            "TimedMetadata": "NONE",
            "VideoPid": 481,
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
              492
            ]
          }
        },
        "VideoDescription": {
          "Width": 640,
          "ScalingBehavior": "DEFAULT",
          "Height": 360,
          "TimecodeInsertion": "DISABLED",
          "AntiAlias": "ENABLED",
          "Sharpness": 50,
          "CodecSettings": {
            "Codec": "H_264",
            "H264Settings": {
              "InterlaceMode": "PROGRESSIVE",
              "NumberReferenceFrames": 3,
              "Syntax": "DEFAULT",
              "Softness": 0,
              "GopClosedCadence": 1,
              "GopSize": 2,
              "Slices": 1,
              "GopBReference": "DISABLED",
              "MaxBitrate": 1200000,
              "SlowPal": "DISABLED",
              "SpatialAdaptiveQuantization": "ENABLED",
              "TemporalAdaptiveQuantization": "ENABLED",
              "FlickerAdaptiveQuantization": "DISABLED",
              "EntropyEncoding": "CABAC",
              "FramerateControl": "INITIALIZE_FROM_SOURCE",
              "RateControlMode": "QVBR",
              "CodecProfile": "MAIN",
```

```
                "Telecine": "NONE",
                "MinIInterval": 0,
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "TRANSITION_DETECTION",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_360"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "TimedMetadataPid": 502,
              "VideoPid": 481,
```

```
            "AudioPids": [
              482,
              483,
              484,
              485,
              486,
              487,
              488,
              489,
              490,
              491,
              492
            ]
          }
        },
        "VideoDescription": {
          "Width": 960,
          "ScalingBehavior": "DEFAULT",
          "Height": 540,
          "TimecodeInsertion": "DISABLED",
          "AntiAlias": "ENABLED",
          "Sharpness": 50,
          "CodecSettings": {
            "Codec": "H_264",
            "H264Settings": {
              "InterlaceMode": "PROGRESSIVE",
              "NumberReferenceFrames": 3,
              "Syntax": "DEFAULT",
              "Softness": 0,
              "GopClosedCadence": 1,
              "GopSize": 2,
              "Slices": 1,
              "GopBReference": "DISABLED",
              "MaxBitrate": 3500000,
              "SlowPal": "DISABLED",
              "SpatialAdaptiveQuantization": "ENABLED",
              "TemporalAdaptiveQuantization": "ENABLED",
              "FlickerAdaptiveQuantization": "DISABLED",
              "EntropyEncoding": "CABAC",
              "FramerateControl": "INITIALIZE_FROM_SOURCE",
              "RateControlMode": "QVBR",
              "CodecProfile": "MAIN",
              "Telecine": "NONE",
              "MinIInterval": 0,
```

```
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "TRANSITION_DETECTION",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_540"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
```

```
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 50,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
                "InterlaceMode": "PROGRESSIVE",
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "GopSize": 2,
                "Slices": 1,
                "GopBReference": "DISABLED",
                "MaxBitrate": 5000000,
                "SlowPal": "DISABLED",
                "SpatialAdaptiveQuantization": "ENABLED",
                "TemporalAdaptiveQuantization": "ENABLED",
                "FlickerAdaptiveQuantization": "DISABLED",
                "EntropyEncoding": "CABAC",
                "FramerateControl": "INITIALIZE_FROM_SOURCE",
                "RateControlMode": "QVBR",
                "CodecProfile": "MAIN",
                "Telecine": "NONE",
                "MinIInterval": 0,
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
```

```
                "SceneChangeDetect": "TRANSITION_DETECTION",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_720"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {}
          },
          "AudioDescriptions": [
            {
              "AudioSourceName": "Audio Selector 1",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "Bitrate": 96000,
                  "CodingMode": "CODING_MODE_2_0",
                  "SampleRate": 48000
                }
              }
            }
          ],
          "OutputSettings": {
```

```
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
            }
          },
          "NameModifier": "_audio"
        }
      ],
      "OutputGroupSettings": {
        "Type": "HLS_GROUP_SETTINGS",
        "HlsGroupSettings": {
          "ManifestDurationFormat": "INTEGER",
          "SegmentLength": 6,
          "TimedMetadataId3Period": 10,
          "CaptionLanguageSetting": "OMIT",
          "Destination": "s3://EXAMPLE-BUCKET/HLS/",
          "DestinationSettings": {
            "S3Settings": {
              "AccessControl": {
                "CannedAcl": "PUBLIC_READ"
              }
            }
          },
          "TimedMetadataId3Frame": "PRIV",
          "CodecSpecification": "RFC_4281",
          "OutputSelection": "MANIFESTS_AND_SEGMENTS",
          "ProgramDateTimePeriod": 600,
          "MinSegmentLength": 0,
          "DirectoryStructure": "SINGLE_DIRECTORY",
          "ProgramDateTime": "EXCLUDE",
          "SegmentControl": "SEGMENTED_FILES",
          "ManifestCompression": "NONE",
          "ClientCache": "ENABLED",
          "StreamInfResolution": "INCLUDE"
        }
      }
    },
    {
      "CustomName": "MP4",
      "Name": "File Group",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "MP4",
```

```
            "Mp4Settings": {
              "CslgAtom": "INCLUDE",
              "FreeSpaceBox": "EXCLUDE",
              "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
            }
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 100,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
                "InterlaceMode": "PROGRESSIVE",
                "ParNumerator": 1,
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "HrdBufferInitialFillPercentage": 90,
                "GopSize": 2,
                "Slices": 2,
                "GopBReference": "ENABLED",
                "HrdBufferSize": 10000000,
                "MaxBitrate": 5000000,
                "ParDenominator": 1,
                "EntropyEncoding": "CABAC",
                "RateControlMode": "QVBR",
                "CodecProfile": "HIGH",
                "MinIInterval": 0,
                "AdaptiveQuantization": "AUTO",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "SPECIFIED",
                "NumberBFramesBetweenReferenceFrames": 3,
                "RepeatPps": "DISABLED",
                "DynamicSubGop": "ADAPTIVE"
```

```
            }
          },
          "AfdSignaling": "NONE",
          "DropFrameTimecode": "ENABLED",
          "RespondToAfd": "NONE",
          "ColorMetadata": "INSERT"
        },
        "AudioDescriptions": [
          {
            "AudioTypeControl": "FOLLOW_INPUT",
            "AudioSourceName": "Audio Selector 1",
            "CodecSettings": {
              "Codec": "AAC",
              "AacSettings": {
                "AudioDescriptionBroadcasterMix": "NORMAL",
                "Bitrate": 160000,
                "RateControlMode": "CBR",
                "CodecProfile": "LC",
                "CodingMode": "CODING_MODE_2_0",
                "RawFormat": "NONE",
                "SampleRate": 48000,
                "Specification": "MPEG4"
              }
            },
            "LanguageCodeControl": "FOLLOW_INPUT",
            "AudioType": 0
          }
        ]
      }
    ],
    "OutputGroupSettings": {
      "Type": "FILE_GROUP_SETTINGS",
      "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/MP4/",
        "DestinationSettings": {
          "S3Settings": {
            "AccessControl": {
              "CannedAcl": "PUBLIC_READ"
            }
          }
        }
      }
    }
  },
```

```
    {
      "CustomName": "Thumbnails",
      "Name": "File Group",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "RAW"
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 50,
            "CodecSettings": {
              "Codec": "FRAME_CAPTURE",
              "FrameCaptureSettings": {
                "FramerateNumerator": 1,
                "FramerateDenominator": 5,
                "MaxCaptures": 500,
                "Quality": 80
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          }
        }
      ],
      "OutputGroupSettings": {
        "Type": "FILE_GROUP_SETTINGS",
        "FileGroupSettings": {
          "Destination": "s3://EXAMPLE-BUCKET/Thumbnails/",
          "DestinationSettings": {
            "S3Settings": {
              "AccessControl": {
                "CannedAcl": "PUBLIC_READ"
              }
            }
          }
        }
      }
```

```
    }
  ],
  "AdAvailOffset": 0,
  "Inputs": [
    {
      "AudioSelectors": {
        "Audio Selector 1": {
          "Offset": 0,
          "DefaultSelection": "DEFAULT",
          "ProgramSelection": 1
        }
      },
      "VideoSelector": {
        "ColorSpace": "FOLLOW"
      },
      "FilterEnable": "AUTO",
      "PsiControl": "USE_PSI",
      "FilterStrength": 0,
      "DeblockFilter": "DISABLED",
      "DenoiseFilter": "DISABLED",
      "TimecodeSource": "EMBEDDED",
      "FileInput": "s3://EXAMPLE-INPUT-BUCKET/input.mp4"
    }
  ]
}
```

3. Dans le dossier `batch-transcode`, créez un fichier avec une fonction Lambda. Vous pouvez utiliser l'exemple Python suivant et nommer le fichier `convert.py`.

   S3 Batch Operations envoie des données propres à la tâche à une fonction Lambda et demande des données de résultat en retour. Pour consulter des exemples de requête et de réponse pour la fonction Lambda, des informations sur les codes de réponse et de résultat, ainsi que des exemples de fonctions Lambda pour S3 Batch Operations, veuillez consulter AWS Lambda Fonction Invoke.

```
import json
import os
from urllib.parse import urlparse
import uuid
import boto3

"""
When you run an S3 Batch Operations job, your job
```

```
invokes this Lambda function. Specifically, the Lambda function is
invoked on each video object listed in the manifest that you specify
for the S3 Batch Operations job in Step 5.

Input parameter "event": The S3 Batch Operations event as a request
                         for the Lambda function.

Input parameter "context": Context about the event.

Output: A result structure that Amazon S3 uses to interpret the result
        of the operation. It is a job response returned back to S3 Batch
 Operations.
"""
def handler(event, context):

    invocation_schema_version = event['invocationSchemaVersion']
    invocation_id = event['invocationId']
    task_id = event['tasks'][0]['taskId']

    source_s3_key = event['tasks'][0]['s3Key']
    source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[-1]
    source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key

    result_list = []
    result_code = 'Succeeded'
    result_string = 'The input video object was converted successfully.'

    # The type of output group determines which media players can play
    # the files transcoded by MediaConvert.
    # For more information, see Creating outputs with AWS Elemental MediaConvert.
    output_group_type_dict = {
        'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
        'FILE_GROUP_SETTINGS': 'FileGroupSettings',
        'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
        'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
        'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
    }

    try:
        job_name = 'Default'
        with open('job.json') as file:
            job_settings = json.load(file)

        job_settings['Inputs'][0]['FileInput'] = source_s3
```

```
        # The path of each output video is constructed based on the values of
        # the attributes in each object of OutputGroups in the job.json file.
        destination_s3 = 's3://{0}/{1}/{2}' \
            .format(os.environ['DestinationBucket'],
                    os.path.splitext(os.path.basename(source_s3_key))[0],
                    os.path.splitext(os.path.basename(job_name))[0])

        for output_group in job_settings['OutputGroups']:
            output_group_type = output_group['OutputGroupSettings']['Type']
            if output_group_type in output_group_type_dict.keys():
                output_group_type = output_group_type_dict[output_group_type]
                output_group['OutputGroupSettings'][output_group_type]
['Destination'] = \
                    "{0}{1}".format(destination_s3,
                                    urlparse(output_group['OutputGroupSettings']
[output_group_type]['Destination']).path)
            else:
                raise ValueError("Exception: Unknown Output Group Type {}."
                                 .format(output_group_type))

        job_metadata_dict = {
            'assetID': str(uuid.uuid4()),
            'application': os.environ['Application'],
            'input': source_s3,
            'settings': job_name
        }

        region = os.environ['AWS_DEFAULT_REGION']
        endpoints = boto3.client('mediaconvert', region_name=region) \
            .describe_endpoints()
        client = boto3.client('mediaconvert', region_name=region,
                              endpoint_url=endpoints['Endpoints'][0]['Url'],
                              verify=False)

        try:
            client.create_job(Role=os.environ['MediaConvertRole'],
                              UserMetadata=job_metadata_dict,
                              Settings=job_settings)
        # You can customize error handling based on different error codes that
        # MediaConvert can return.
        # For more information, see MediaConvert error codes.
        # When the result_code is TemporaryFailure, S3 Batch Operations retries
        # the task before the job is completed. If this is the final retry,
```

```
        # the error message is included in the final report.
        except Exception as error:
            result_code = 'TemporaryFailure'
            raise

    except Exception as error:
        if result_code != 'TemporaryFailure':
            result_code = 'PermanentFailure'
        result_string = str(error)

    finally:
        result_list.append({
            'taskId': task_id,
            'resultCode': result_code,
            'resultString': result_string,
        })

    return {
        'invocationSchemaVersion': invocation_schema_version,
        'treatMissingKeyAs': 'PermanentFailure',
        'invocationId': invocation_id,
        'results': result_list
    }
```

4. Pour créer un package de déploiement avec `convert.py` et `job.json` sous forme de fichier `.zip` nommé `lambda.zip`, dans votre terminal local, ouvrez le dossier `batch-transcode` que vous avez créé précédemment et exécutez la commande suivante.

   Pour les utilisateurs macOS, exécutez la commande suivante :

   ```
   zip -r lambda.zip convert.py job.json
   ```

   Pour les utilisateurs Windows, exécutez les commandes suivantes :

   ```
   powershell Compress-Archive convert.py lambda.zip
   ```

   ```
   powershell Compress-Archive -update job.json lambda.zip
   ```

Créer une fonction Lambda à l'aide d'un rôle d'exécution (console)

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Dans le panneau de navigation de gauche, choisissez Fonctions.
3. Choisissez Créer une fonction.
4. Choisissez Créer à partir de zéro.
5. Sous Informations de base, procédez comme suit :
   a. Sous Nom de la fonction, saisissez **`tutorial-lambda-convert`**.
   b. Pour Runtime (Exécution), choisissez Python 3.8 ou version ultérieure de Python.
6. Choisissez Modifier le rôle d'exécution par défaut et, sous Rôle d'exécution, choisissez Utiliser un rôle existant.
7. Sous Existing role (Rôle existant), choisissez le nom du rôle IAM que vous avez créé pour votre fonction Lambda à l'étape 3(par exemple, **`tutorial-lambda-transcode-role`**).
8. Pour les paramètres restants, conservez les paramètres par défaut.
9. Choisissez Créer une fonction.

Déployez votre fonction Lambda avec les archives de fichiers .zip, puis configurez la fonction Lambda (console)

1. Dans la section Code Source (Source du code) de la page de la fonction Lambda que vous avez créée (par exemple, **`tutorial-lambda-convert`**), choisissez Upload from (Télécharger à partir de), puis le fichier .zip.
2. Choisissez Charger pour sélectionner votre fichier `.zip` local.
3. Choisissez le fichier `lambda.zip` que vous avez créé précédemment, puis choisissez Open (Ouvrir).
4. Choisissez Enregistrer.
5. Dans la section Paramètres d'exécution, choisissez Modifier.
6. Pour indiquer à l'exécution Lambda la méthode de gestionnaire dans votre code de fonction Lambda à appeler, saisissez **`convert.handler`** dans le champ Handler (Gestionnaire).

   Quand vous configurez une fonction dans Python, la valeur du paramètre de gestionnaire correspond au nom de fichier et au nom de module de gestionnaire, séparés par un point (`.`).

Par exemple, `convert.handler` appelle la méthode `handler` définie dans le fichier `convert.py`.

7. Choisissez Enregistrer.
8. Dans la page de la fonction Lambda, choisissez l'onglet Configuration. Dans le panneau de navigation de gauche de l'onglet Configuration, choisissez Environment variables (Variables d'environnement), puis Edit (Modifier).
9. Choisissez Ajouter une variable d'environnement. Ensuite, saisissez la clé et la valeur spécifiées pour chacune des variables d'environnement suivantes :

   - Clé : Valeur **`DestinationBucket`** : **`tutorial-bucket-2`**

     Cette valeur est le compartiment S3 pour les fichiers multimédias de sortie que vous avez créés à l'étape 1.

   - Clé : Valeur **`MediaConvertRole`** : **`arn:aws:iam::`*`111122223333`*`:role/tutorial-mediaconvert-role`**

     Cette valeur est l'ARN du rôle IAM MediaConvert que vous avez créé à l'étape 2. Assurez-vous de remplacer cet ARN par le véritable ARN de votre rôle IAM.

   - Clé : Valeur **`Application`** : **`Batch-Transcoding`**

     Cette valeur est le nom de l'application.

10. Choisissez Enregistrer.
11. (Facultatif) Dans l'onglet Configuration, dans la section Configuration générale du panneau de navigation de gauche, choisissez Modifier. Dans le champ Délai d'expiration, saisissez **`2`** min **`0`** sec. Ensuite, choisissez Enregistrer.

    Le Délai d'expiration est le temps que Lambda autorise pour l'exécution d'une fonction avant de l'arrêter. Le durée par défaut est de 3 secondes. La tarification est basée sur la quantité de mémoire configurée et la durée pendant laquelle votre code s'exécute. Pour en savoir plus, consultez AWS Lambda Tarification.

Étape 5 : Configurer un inventaire Amazon S3 pour votre compartiment source S3

Après avoir configuré la fonction Lambda de transcodage, créez une tâche S3 Batch Operations pour transcoder un jeu de vidéos. Tout d'abord, vous avez besoin d'une liste d'objets vidéo d'entrée sur

lesquels S3 Batch Operations devra exécuter l'action de transcodage indiquée. Pour obtenir une liste d'objets vidéo d'entrée, vous pouvez générer un rapport d'inventaire S3 pour votre compartiment source S3 (par exemple, **`tutorial-bucket-1`**).

Sous-étapes

- Créer et configurer un compartiment pour les rapports d'inventaire S3 des vidéos d'entrée
- Configurer l'inventaire Amazon S3 pour votre compartiment source S3 des vidéos
- Consultez le rapport d'inventaire pour votre compartiment source S3 de vidéos

Créer et configurer un compartiment pour les rapports d'inventaire S3 des vidéos d'entrée

Pour stocker un rapport d'inventaire S3 qui répertorie les objets du compartiment source S3, créez un compartiment de destination d'inventaire S3, puis configurez une politique de compartiment pour que le compartiment puisse écrire des fichiers d'inventaire dans le compartiment source S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.
4. Pour Bucket Name (Nom du compartiment), indiquez le nom de votre compartiment, (par exemple, **`tutorial-bucket-3`**).
5. Pour Région AWS, choisissez l' Région AWS endroit où vous souhaitez que le compartiment réside.

   Le compartiment de destination de l'inventaire doit se trouver dans le même compartiment Région AWS que le compartiment source dans lequel vous configurez S3 Inventory. Le compartiment de destination d'inventaire peut être dans un autre Compte AWS.
6. Dans Block Public Access settings for this bucket (Bloquer les paramètres d'accès public pour ce compartiment), conservez les paramètres par défaut (Block all public access Bloquer tous les accès publics) est activé).
7. Pour les paramètres restants, conservez les paramètres par défaut.
8. Choisissez Créer un compartiment.
9. Dans la liste Compartiments, choisissez le nom du compartiment que vous venez de créer (par exemple, **`tutorial-bucket-3`**).

10. Pour accorder à Amazon S3 l'autorisation d'écrire des données pour les rapports d'inventaire dans le compartiment de destination d'inventaire S3, choisissez l'onglet Permissions (Autorisations).
11. Faites défiler jusqu'à la section Bucket policy (Stratégie de compartiment), puis choisissez Edit (Modifier). La page Bucket policy (Stratégie de compartiment) s'ouvre.
12. Pour accorder des autorisations pour l'inventaire S3, dans le champ Policy (Stratégie), collez la stratégie de compartiment suivante.

    Remplacez les trois exemples de valeurs par les valeurs suivantes :

    - Le nom du compartiment que vous avez créé pour stocker les rapports d'inventaire (par exemple, *tutorial-bucket-3*).
    - Le nom du compartiment source qui stocke les vidéos d'entrée (par exemple, *tutorial-bucket-1*).
    - L' Compte AWS ID que vous avez utilisé pour créer le compartiment de source vidéo S3 (par exemple,*111122223333*.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"InventoryAndAnalyticsExamplePolicy",
      "Effect":"Allow",
      "Principal": {"Service": "s3.amazonaws.com"},
      "Action":"s3:PutObject",
      "Resource":["arn:aws:s3:::tutorial-bucket-3/*"],
      "Condition": {
          "ArnLike": {
              "aws:SourceArn": "arn:aws:s3:::tutorial-bucket-1"
           },
         "StringEquals": {
             "aws:SourceAccount": "111122223333",
             "s3:x-amz-acl": "bucket-owner-full-control"
          }
       }
    }
  ]
}
```

13. Choisissez Enregistrer les modifications.

Configurer l'inventaire Amazon S3 pour votre compartiment source S3 des vidéos

Pour générer une liste de fichiers plats d'objets vidéo et de métadonnées, vous devez configurer l'inventaire S3 pour votre compartiment source S3 des vidéos. Ces rapports d'inventaire planifiés peuvent inclure tous les objets du compartiment ou les objets regroupés par un préfixe partagé. Dans ce didacticiel, le rapport d'inventaire S3 inclut tous les objets vidéo de votre compartiment source S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Pour configurer un rapport d'inventaire S3 des vidéos d'entrée dans votre compartiment source S3, dans la liste Buckets (Compartiments), choisissez le nom du compartiment source S3 (par exemple, **`tutorial-bucket-1`**).
4. Choisissez l'onglet Gestion.
5. Faites défiler jusqu'à la section Inventory configurations (Configurations d'inventaire) et choisissez Create inventory configuration (Créer une configuration d'inventaire).
6. Pour le Nom de la configuration d'inventaire, saisissez un nom (par exemple, **`tutorial-inventory-config`**).
7. Sous Inventory scope (Portée de l'inventaire), choisissez Current version only (Version actuelle uniquement pour Object versions (Versions d'un objet) et conservez les autres paramètres Inventory scope (Portée de l'inventaire) définis pour les valeurs par défaut de ce didacticiel.
8. Dans la section Report details (Détails du rapport), pour Destination bucket (Compartiment de destination), choisissez This account (Ce compte).
9. Pour Destination, choisissez Browse S3 (Parcourir S3), puis choisissez le compartiment de destination que vous avez créé précédemment pour enregistrer les rapports d'inventaire (par exemple, **`tutorial-bucket-3`**). Ensuite, choisissez Choose path (Choisir un chemin).

   Le compartiment de destination de l'inventaire doit se trouver dans le même compartiment Région AWS que le compartiment source dans lequel vous configurez S3 Inventory. Le compartiment de destination d'inventaire peut être dans un autre Compte AWS.

   Sous le champ du compartiment Destination, l'autorisation du compartiment de destination est ajoutée à la politique du compartiment de destination afin de permettre à Amazon S3 de

placer des données dans le compartiment de destination d'inventaire. Pour plus d'informations, consultez Création d'une stratégie de compartiment de destination.

10. Pour Frequency (Fréquence), choisissez Daily (Quotidiennement).
11. Pour Output format (Format de sortie), choisissez CSV.
12. Pour Status (Statut), choisissez Enabled (Activé).
13. Dans la section Server-side encryption (Chiffrement côté serveur), choisissez Disable (Désactiver) pour ce didacticiel.

    Pour plus d'informations, consultez Configuration de l'inventaire à l'aide de la console S3 et Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement.
14. Dans la section Additional fields - optional (Champs supplémentaires – facultatif), sélectionnez Size (Taille) Last modified (Dernière modification) et Storage class (Classe de stockage).
15. Sélectionnez Créer.

Pour plus d'informations, consultez Configuration de l'inventaire à l'aide de la console S3.

Consultez le rapport d'inventaire pour votre compartiment source S3 de vidéos

Lorsqu'un rapport d'inventaire est publié, les fichiers manifestes sont envoyés au compartiment de destination d'inventaire S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste des Compartiments, choisissez le nom du compartiment source des vidéos (par exemple, **`tutorial-bucket-1`**).
4. Choisissez Gestion.
5. Pour savoir si votre rapport d'inventaire S3 est prêt afin que vous puissiez créer une tâche S3 Batch Operations à l'étape 7, sous Inventory configurations (Configurations d'inventaire), vérifiez si le bouton Create job from manifest (Créer une tâche à partir du manifeste) est activé.

> **Note**
>
> La distribution du premier rapport d'inventaire peut prendre jusqu'à 48 heures. Si l'icône Créer une tâche à partir du manifeste est désactivé, cela signifie que le premier rapport d'inventaire n'a pas été remis. Attendez que le premier rapport d'inventaire soit livré et

que le bouton Create job from manifest (Créer une tâche à partir du manifeste) soit activé pour créer une tâche S3 Batch Operations à l'étape 7.

6. Pour vérifier un rapport d'inventaire S3 (`manifest.json`), dans la colonne Destination, choisissez le nom du compartiment de destination d'inventaire que vous avez créé précédemment pour stocker les rapports d'inventaire (par exemple, **`tutorial-bucket-3`**).
7. Dans l'onglet Objects (Objets), choisissez le dossier existant portant le nom de votre compartiment source S3 (par exemple, **`tutorial-bucket-1`**). Choisissez ensuite le nom dans lequel vous avez saisi dans Inventory configuration name (Nom de configuration d'inventaire) lorsque vous avez créé la configuration d'inventaire précédemment (par exemple, **`tutorial-inventory-config`**).

   Vous pouvez afficher une liste de dossiers avec les dates de génération des rapports en guise de noms.
8. Pour vérifier le rapport d'inventaire quotidien S3 à une date donnée, choisissez le dossier avec le nom de date de génération correspondant, puis choisissez `manifest.json`.
9. Pour vérifier les détails du rapport d'inventaire à une date précise, dans la page manifest.json, choisissez Télécharger ou Ouvrir.

Étape 6 : Créer un rôle IAM pour S3 Batch Operations

Pour utiliser S3 Batch Operations à des fins de transcodage par lots, vous devez d'abord créer un rôle IAM pour permettre à Amazon S3 d'exécuter S3 Batch Operations.

Sous-étapes

- Créer une politique IAM pour S3 Batch Operations
- Créez un rôle IAM S3 Batch Operations et attachez les politiques d'autorisations

Créer une politique IAM pour S3 Batch Operations

Vous devez créer une stratégie IAM qui autorise lS3 Batch Operations à lire le manifeste d'entrée, appeler la fonction Lambda et écrire le rapport de fin de tâche S3 Batch Operations.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Politiques.

3. Sélectionnez Créer une politique.
4. Choisissez l'onglet JSON.
5. Dans le champ JSON, collez la politique JSON suivante.

   Dans la stratégie JSON, remplacez les quatre valeurs d'exemple par les valeurs suivantes :

   - Le nom du compartiment source qui stocke vos vidéos d'entrée (par exemple, *tutorial-bucket-1*).
   - Le nom du compartiment de destination d'inventaire que vous avez créé à l'étape 5 pour stocker des fichiers `manifest.json` (par exemple, *tutorial-bucket-3*).
   - Le nom du compartiment que vous avez créé à l'étape 1 pour stocker des fichiers multimédias de sortie (par exemple, *tutorial-bucket-2*). Dans ce didacticiel, nous mettons les rapports de fin de tâche dans le compartiment de destination des fichiers multimédias de sortie.
   - L'ARN du rôle de la fonction Lambda que vous avez créée à l'étape 4. Pour rechercher et copier l'ARN de rôle de la fonction Lambda, procédez comme suit :
     - Dans un nouvel onglet du navigateur, ouvrez la page Functions (Fonctions) de la console Lambda à l'adresse https://console.aws.amazon.com/lambda/home#/functions.
     - Dans la liste Functions (Fonctions), choisissez le nom de la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **tutorial-lambda-convert**).
     - Choisissez Copier l'ARN.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3Get",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::tutorial-bucket-1/*",
                "arn:aws:s3:::tutorial-bucket-3/*"
            ]
        },
        {
            "Sid": "S3PutJobCompletionReport",
```

```
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::tutorial-bucket-2/*"
        },
        {
            "Sid": "S3BatchOperationsInvokeLambda",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:us-west-2:111122223333:function:tutorial-lambda-
convert"
            ]
        }
    ]
}
```

6. Choisissez Étape suivante : balises.
7. Choisissez Étape suivante : vérification).
8. Dans le champ Nom, saisissez **`tutorial-s3batch-policy`**.
9. Choisissez Créer une politique.

Créez un rôle IAM S3 Batch Operations et attachez les politiques d'autorisations

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Choisissez le type de rôle Service AWS, puis le service S3.
4. Sous Sélectionner votre cas d'utilisation, choisissez S3 Batch Operations.
5. Choisissez Étape suivante : autorisations.
6. Sous Attach permissions policies (Attacher des stratégies d'autorisation), saisissez le nom de la stratégie IAM que vous avez créée précédemment (par exemple, **`tutorial-s3batch-policy`**) dans la zone de recherche pour filtrer la liste de stratégies. Cochez la case en regard du nom de la stratégie (par exemple, **`tutorial-s3batch-policy`**).
7. Choisissez Suivant : Balises.

8. Choisissez Next: Review (Suivant : Vérification).
9. Pour le Nom du rôle, saisissez **`tutorial-s3batch-role`**.
10. Choisissez Créer un rôle.

    Après avoir créé le rôle IAM pour S3 Batch Operations, la politique de confiance suivante sera automatiquement attachée au rôle. Cette stratégie d'approbation permet au principal du service S3 Batch Operations d'assumer le rôle IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Étape 7 : Configurer et exécuter une tâche S3 Batch Operations

Pour créer une tâche S3 Batch Operations aux fins de traiter les vidéos d'entrée dans votre compartiment source S3, vous devez préciser des paramètres pour cette tâche particulière.

> **Note**
>
> Avant de commencer à créer une tâche S3 Batch Operations, vérifiez que le bouton Create job from manifest (Créer une tâche à partir du manifeste) est activé. Pour plus d'informations, consultez Consultez le rapport d'inventaire pour votre compartiment source S3 de vidéos. Si le bouton Create job from manifest (Créer une tâche à partir du manifeste) est désactivé, cela signifie que le premier rapport d'inventaire n'a pas été livré et vous devez attendre que le bouton soit activé. Après avoir configuré l'inventaire Amazon S3 pour votre compartiment source S3 à l'étape 5, la livraison du premier rapport d'inventaire peut prendre jusqu'à 48 heures.

Sous-étapes

- Créer une tâche S3 Batch Operations
- Exécutez la tâche S3 Batch Operations pour appeler votre fonction Lambda
- (Facultatif) Vérifiez votre rapport d'achèvement
- (Facultatif) Contrôlez chaque appel Lambda dans la console Lambda
- (Facultatif) Surveillez chaque tâche de MediaConvert transcodage vidéo dans la console MediaConvert

Créer une tâche S3 Batch Operations

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Choisissez Créer une tâche.
4. Pour Région AWS, choisissez la Région dans laquelle vous souhaitez créer votre tâche.

   Dans ce didacticiel, pour utiliser la tâche S3 Batch Operations aux fins d'appeler une fonction Lambda, vous devez créer la tâche dans la même Région que le compartiment de source S3 des vidéos où se trouvent les objets référencés dans le manifeste.
5. Dans la section Manifest (Manifeste), procédez comme suit :

   a. Pour Manifest format (Format du manifeste), choisissez S3 Inventory report (manifest.json) (Rapport d'inventaire S3 [manifest.json]).

   b. Pour Manifest object (Objet du manifeste), choisissez Browse S3 (Parcourir S3) pour chercher le compartiment que vous avez créé à l'étape 5 pour stocker des rapports d'inventaire (par exemple, **`tutorial-bucket-3`**). Sur la page Manifest object (Objet manifeste), parcourez les noms des objets jusqu'à ce que vous trouviez un fichier `manifest.json` pour une date spécifique. Ce fichier répertorie les informations sur toutes les vidéos que vous souhaitez transcoder par lots. Lorsque vous avez trouvé le fichier `manifest.json` que vous souhaitez utiliser, choisissez le bouton d'option en regard de celui-ci. Ensuite, choisissez Choose path (Choisir un chemin).

   c. (Facultatif) Pour Manifest object version ID - optional (ID de version de l'objet manifeste) - facultatif), saisissez l'ID de version de l'objet manifeste si vous souhaitez utiliser une autre version que la plus récente.
6. Choisissez Suivant.

7. Pour utiliser la fonction Lambda afin de transcoder tous les objets répertoriés dans le fichier `manifest.json` sélectionné, sous Operation type (Type d'opération), choisissez Invoke AWS Lambda function (Appeler une fonction Lambda).
8. Dans la section Invoke Lambda function (Appeler une fonction Lambda), procédez comme suit :

   a. Choisissez Choisir parmi les fonctions dans votre compte.

   b. Pour Lambda function (Fonction Lambda), choisissez la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **`tutorial-lambda-convert`**).

   c. Pour Lambda function version (Version de la fonction Lambda), conservez la valeur par défaut $LATEST.
9. Choisissez Suivant. La page Configure additional options (Configurer des options supplémentaires) s'ouvre.
10. Dans la section Additional options (Options supplémentaires), conservez les paramètres par défaut.

    Pour plus d'informations sur ces options, consultez Éléments d'une demande de tâche d'opération par lot.
11. Dans la section Completion report (Rapport d'achèvement), pour Path to completion report destination (Chemin vers la destination du rapport d'achèvement), choisissez Browse S3 (Parcourir S3). Recherchez le compartiment que vous avez créé à l'étape 1 pour des fichiers multimédias de sortie (par exemple, **`tutorial-bucket-2`**). Choisissez le bouton d'option en regard du nom de ce compartiment. Ensuite, choisissez Choose path (Choisir un chemin).

    Pour les paramètres de rapport d'achèvement restants, conservez les paramètres par défaut. Pour en savoir plus sur les paramètres des rapports d'achèvement, consultez Éléments d'une demande de tâche d'opération par lot. Un rapport d'achèvement conserve un enregistrement des détails de la tâche et des opérations exécutées.
12. Dans la section Permissions (Autorisations), choisissez Choose from existing IAM roles (Choisir parmi les rôles IAM existants). Pour Rôle IAM, choisissez le rôle IAM pour votre tâche S3 Batch Operations job que vous avez créée à l'étape 6 (par exemple, **`tutorial-s3batch-role`**).
13. Choisissez Suivant.
14. Sur la page Review (Vérification), vérifiez les paramètres. Ensuite, choisissez Create job (Créer une tâche).

    Lorsque S3 termine la lecture du manifeste de votre tâche S3 Batch Operations, la tâche passe à l'état Awaiting your confirmation to run (En attente de confirmation). Pour afficher les mises à jour

de l'état de la tâche, actualisez la page. Vous ne pouvez pas exécuter votre tâche tant que son statut n'est pas Awaiting your confirmation to run (En attente de confirmation).

Exécutez la tâche S3 Batch Operations pour appeler votre fonction Lambda

Exécutez votre tâche d'opérations par lots pour appeler votre fonction Lambda aux fins de transcodage vidéo. Si votre tâche échoue, vous pouvez vérifier votre rapport d'achèvement pour identifier la cause.

Pour exécuter la tâche S3 Batch Operations

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Dans la liste Jobs (Tâches), choisissez l'ID de tâche de la tâche sur la première ligne, qui est la tâche S3 Batch Operations que vous avez créée précédemment.
4. Choisissez Exécuter la tâche.
5. Vérifiez à nouveau les paramètres de votre tâche et confirmez que la valeur du Nombre total d'objets répertoriés dans le manifeste est le même que le nombre d'objets indiqué dans le manifeste. Ensuite, choisissez Run job (Exécuter la tâche).

   La page de votre tâche S3 Batch Operations s'ouvre.
6. Lorsque la tâche aura commencé à s'exécuter, dans la page de votre tâche, sous État, vérifiez la progression de votre tâche S3 Batch Operations, par exemple État, % Terminé, Total réussi (débit),Total échoué (débit), Date de la résiliation, et Raison de la résiliation.

   Lorsque la tâche S3 Batch Operations sera terminée, affichez les données dans la page de votre tâche pour confirmer qu'elle s'est achevée comme prévu.

   Si plus de 50 % des opérations objet d'une tâche S3 Batch Operations échouent après que plus de 1 000 opérations aient été tentées, la tâche échouera automatiquement. Pour vérifier votre rapport d'achèvement afin d'identifier la cause des échecs, utilisez la procédure facultative ci-dessous.

(Facultatif) Vérifiez votre rapport d'achèvement

Vous pouvez utiliser votre rapport d'achèvement pour déterminer quels objets ont échoué et la cause des échecs.

Pour vérifier votre rapport d'achèvement afin d'obtenir plus de détails sur les objets qui ont échoué

1. Dans la page de votre tâche S3 Batch Operations, faites défiler jusqu'à la section Completion report (Rapport d'achèvement), puis cliquez sur le lien sous Completion report destination (Destination du rapport d'achèvement).

   La page du compartiment de destination de sortie S3 s'ouvre.
2. Dans l'onglet Objects (Objets), choisissez le dossier dont le nom se termine par l'ID de tâche de la tâche S3 Batch Operations que vous avez créée précédemment.
3. Choisissez résultats/.
4. Cochez la case en regard du fichier `.csv`.
5. Pour afficher le rapport de la tâche, choisissez Open (Ouvrir) ou Download (Télécharger).

(Facultatif) Contrôlez chaque appel Lambda dans la console Lambda

Lorsque la tâche S3 Batch Operations commence à s'exécuter, la tâche appelle la fonction Lambda pour chaque objet d'entrée vidéo. S3 écrit les journaux de chaque appel Lambda dans Logs. CloudWatch Vous pouvez utiliser le tableau de bord de surveillance de la console Lambda pour surveiller votre fonction Lambda.

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Dans le panneau de navigation de gauche, choisissez Fonctions.
3. Dans la liste Functions (Fonctions), choisissez le nom de la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **`tutorial-lambda-convert`**).
4. Choisissez l'onglet Surveiller.
5. Sous Métriques, consultez les mesures de l'environnement d'exécution de votre fonction Lambda.
6. Sous Logs, consultez les données du journal pour chaque appel Lambda via CloudWatch Logs Insights.

> **Note**
>
> Si vous utilisez des opérations de lots S3 avec une fonction Lambda, la fonction Lambda sera appelée sur chaque objet. Si votre tâche S3 Batch Operations est volumineuse, elle pourra appeler plusieurs fonctions Lambda en même temps, provoquant un pic dans la simultanéité Lambda.
> Chacune Compte AWS dispose d'un quota de simultanéité Lambda par région. Pour plus d'informations, veuillez consulter AWS Lambda Function Scaling (Mise à l'échelle de fonction Lambda) dans le Guide du développeur AWS Lambda . Une bonne pratique pour utiliser les fonctions Lambda avec les opérations par lots S3 consiste à définir une limite de simultanéité sur la fonction Lambda elle-même. La définition d'une limite de simultanéité empêche votre tâche de consommer la majeure partie de votre simultanéité Lambda et de limiter éventuellement d'autres fonctions de votre compte. Pour plus d'informations, consultez Gestion de la simultanéité réservée Lambda dans le Guide du développeur AWS Lambda .

(Facultatif) Surveillez chaque tâche de MediaConvert transcodage vidéo dans la console MediaConvert

Une MediaConvert tâche consiste à transcoder un fichier multimédia. Lorsque votre tâche S3 Batch Operations appelle votre fonction Lambda pour chaque vidéo, chaque appel de fonction Lambda crée MediaConvert une tâche de transcodage pour chaque vidéo en entrée.

1. Connectez-vous à la MediaConvert console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/mediaconvert/.
2. Si la page MediaConvert d'introduction apparaît, choisissez Commencer.
3. Dans la liste des Tâches, affichez chaque ligne pour contrôler la tâche de transcodage pour chaque vidéo d'entrée.
4. Identifiez la ligne d'une tâche à vérifier, puis cliquez sur le lien Job ID (ID de tâche) pour ouvrir la page de détails de la tâche.
5. Dans la page Résumé de la tâche, sous Sorties, cliquez sur le lien de la sortie HLS, MP4 ou Miniatures, en fonction de ce qui est pris en charge par votre navigateur, pour accéder au compartiment de destination S3 des fichiers multimédias de sortie.

6. Dans le dossier correspondant (HLS, MP4 ou Miniatures) de votre compartiment de destination de sortie S3, choisissez le nom de l'objet fichier multimédia de sortie.

   La page de détails de l'objet s'ouvre.

7. Dans la page de détails de l'objet, sous Object overview (Présentation de l'objet), choisissez le lien sous la rubrique Object URL (URL de l'objet) pour regarder le fichier multimédia de sortie transcodé.

Étape 8 : Vérifier les fichiers multimédias de sortie à partir de votre compartiment de destination S3

Vérifier les fichiers multimédias de sortie à partir de votre compartiment de destination S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment de destination S3 pour les fichiers multimédias de sortie que vous avez créés à l'étape 1 (par exemple, **`tutorial-bucket-2`**).
4. Dans l'onglet Objets, chaque vidéo d'entrée possède un dossier portant le nom de la vidéo d'entrée. Chaque dossier contient les fichiers multimédias de sortie transcodés d'une vidéo d'entrée.

   Pour vérifier les fichiers multimédias de sortie d'une vidéo d'entrée, procédez comme suit :

   a. Choisissez le dossier portant le nom de la vidéo d'entrée que vous souhaitez vérifier.
   b. Choisissez le dossier Par défaut/.
   c. Choisissez le dossier d'un format transcodé (HLS, MP4 ou vignettes dans ce didacticiel).
   d. Choisissez le nom du fichier multimédia de sortie.
   e. Pour regarder le fichier transcodé, choisissez le lien sous URL de l'objet dans la page de détails de l'objet.

      Les fichiers multimédia de sortie au format HLS sont divisés en segments courts. Pour lire ces vidéos, intégrez l'URL de l'objet du fichier `.m3u8` dans un lecteur compatible.

Étape 9 : Nettoyer

Si vous avez transcodé des vidéos à l'aide de S3 Batch Operations, Lambda, MediaConvert et uniquement dans le cadre d'un exercice d'apprentissage, supprimez AWS les ressources que vous avez allouées afin de ne plus payer de frais.

Sous-étapes

- Supprimer la configuration d'inventaire S3 de votre compartiment source S3
- Supprimer la fonction Lambda
- Supprimer le groupe de CloudWatch journaux
- Supprimer les rôles IAM en même temps que les politiques en ligne des rôles IAM
- Supprimer la stratégie IAM gérée par le client
- Vider les compartiments S3
- Supprimer les compartiments S3

Supprimer la configuration d'inventaire S3 de votre compartiment source S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans Compartiments, choisissez le nom de votre compartiment source (par exemple, **`tutorial-bucket-1`**).
4. Choisissez l'onglet Gestion.
5. Sous Inventory configurations (Configurations d'inventaire), choisissez le bouton radio en regard de la configuration d'inventaire que vous avez créée à l'étape 5 (par exemple, **`tutorial-inventory-config`**).
6. Choisissez Supprimer, puis choisissez Confirmer.

Supprimer la fonction Lambda

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Dans le panneau de navigation de gauche, choisissez Fonctions.

3. Cochez la case en regard de la fonction que vous avez créée à l'étape 4 (par exemple, **`tutorial-lambda-convert`**).
4. Choisissez Actions, puis Supprimer.
5. Dans la boîte de dialogue Supprimer une fonction, choisissez Supprimer.

Supprimer le groupe de CloudWatch journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Logs (Journaux), puis Log groups (Groupes de journaux).
3. Cochez la case en regard du groupe de journaux dont le nom se termine par la fonction Lambda que vous avez créée à l'étape 4 (par exemple, **`tutorial-lambda-convert`**).
4. Choisissez Actions, puis Supprimer le ou les groupes de journaux.
5. Dans la boîte de dialogue Supprimer le ou les groupes de journaux), choisissez Supprimer.

Supprimer les rôles IAM en même temps que les politiques en ligne des rôles IAM

Pour supprimer les rôles IAM que vous avez créés à l'étape 2, à l'étape 3 et à l'étape 6, procédez comme suit :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles), puis cochez les cases en regard des noms que vous souhaitez supprimer.
3. En haut de la page, choisissez Supprimer.
4. Dans la boîte de dialogue de confirmation, saisissez la réponse requise dans la zone de saisie de texte en fonction de l'invite, puis choisissez Supprimer.

Supprimer la stratégie IAM gérée par le client

Pour supprimer la stratégie IAM gérée par le client que vous avez créée à l'étape 6, procédez comme suit :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Choisissez le bouton radio en regard de la stratégie que vous avez créée à l'étape 6 (par exemple, **`tutorial-s3batch-policy`**). Vous pouvez utiliser la zone de recherche pour filtrer la liste des politiques.
4. Choisissez Actions, puis choisissez Supprimer.
5. Confirmez que vous souhaitez supprimer cette stratégie en saisissant son nom dans le champ de texte, puis choisissez Delete (Supprimer).

Vider les compartiments S3

Pour vider les compartiments S3 que vous avez créés dans Prerequisites (Prérequis), à l'étape 1 et à l'étape 5, procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le bouton radio en regard du nom du compartiment à vider, puis choisissez Empty (Vider).
4. Dans la page Empty bucket (Vider le compartiment), confirmez que vous souhaitez vider le compartiment en saisissant **`permanently delete`** dans le champ de texte, puis choisissez Empty (Vider).

Supprimer les compartiments S3

Pour supprimer les compartiments S3 que vous avez créés dans Prerequisites (Prérequis), à l'étape 1 et à l'étape 5, procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le bouton radio en regard du nom du compartiment que vous souhaitez supprimer.
4. Choisissez Supprimer.

5. Dans la page Supprimer le compartiment, confirmez que vous souhaitez supprimer le compartiment en saisissant le nom de ce dernier dans le champ de texte, puis choisissez Supprimer le compartiment.

Étapes suivantes

Après avoir terminé ce didacticiel, vous pourrez explorer d'autres cas d'utilisation pertinents :

- Vous pouvez utiliser Amazon CloudFront pour diffuser les fichiers multimédia transcodés aux spectateurs du monde entier. Pour plus d'informations, consultez Tutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53.
- Vous pouvez transcoder des vidéos au moment où vous les téléchargez dans le compartiment source S3. Pour ce faire, vous pouvez configurer un déclencheur d'événement Amazon S3 qui invoque automatiquement la fonction Lambda pour transcoder de nouveaux objets dans S3 avec. MediaConvert Pour en savoir plus, veuillez consulter Didacticiels : Utilisation d'un déclencheur Amazon S3 pour appeler une fonction Lambda dans le Guide du développeur AWS Lambda ..

Didacticiel : configuration d'un site web statique sur Amazon S3

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Vous pouvez configurer un compartiment Amazon S3 afin qu'il fonctionne comme un site web. Cet exemple explique les différentes étapes d'hébergement d'un site web sur Amazon S3.

> **Important**
>
> Le didacticiel suivant nécessite la désactivation du blocage de l'accès public. Nous vous recommandons de maintenir activé le blocage de l'accès public. Si vous souhaitez conserver les quatre paramètres de blocage de l'accès public activés et héberger un site Web statique, vous pouvez utiliser le contrôle CloudFront d'accès d'origine (OAC) d'Amazon. Amazon CloudFront fournit les fonctionnalités requises pour configurer un site Web statique sécurisé. Les sites Web statiques Amazon S3 prennent uniquement en charge les points de terminaison HTTP. Amazon CloudFront utilise le stockage durable d'Amazon S3 tout en fournissant des en-têtes de sécurité supplémentaires, tels que HTTPS. HTTPS accroît la sécurité en chiffrant une demande HTTP normale et en offrant une protection contre les cyberattaques courantes. Pour plus d'informations, consultez Getting started with a secure static website in the Amazon CloudFront Developer Guide.

Rubriques

Étape 1 : Créer un compartiment

Les instructions ci-dessous fournissent une vue d'ensemble de la façon de créer vos compartiments pour l'hébergement de site web. Pour obtenir des step-by-step instructions détaillées sur la création d'un bucket, consultezCréer un compartiment.

Pour créer un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Créer un compartiment.
3. Saisissez le Nom du compartiment (par exemple **`example.com`**).
4. Choisissez la Région dans laquelle vous souhaitez créer le compartiment.

   Choisissez une Région proche de vous sur le plan géographique afin de limiter la latence et les coûts, ou de répondre aux exigences réglementaires. C'est la Région que vous choisissez qui détermine votre point de terminaison de site web Amazon S3. Pour plus d'informations, consultez Points de terminaison de sites web.
5. Pour accepter les paramètres par défaut et créer le compartiment, choisissez Créer.

Étape 2 : Activer l'hébergement de site web statique

Après avoir créé un compartiment, vous pouvez activer l'hébergement de site web statique pour votre compartiment. Vous pouvez créer un compartiment ou utiliser un compartiment existant.

Pour activer l'hébergement de site Web statique

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer l'hébergement de sites web statiques.
3. Choisissez Propriétés.
4. Sous Static website hosting (Hébergement de site Web statique), choisissez Edit (Modifier).
5. Choisissez Utiliser ce compartiment pour héberger un site Web.
6. Sous Static website hosting (Hébergement de site web statique), choisissez Enable (Activer).
7. Dans Index document (Document d'index), entrez le nom du document d'index, généralement `index.html`.

   Le nom du document d'index est sensible à la casse et doit correspondre exactement au nom de fichier du document d'index HTML que vous prévoyez de charger dans votre compartiment S3. Lorsque vous configurez un compartiment pour l'hébergement d'un site web, vous devez indiquer un document d'index. Amazon S3 renvoie ce document d'index lorsque des demandes sont

faites dans le domaine racine ou dans n'importe quel sous-dossier. Pour plus d'informations, consultez Configuration d'un document d'index.

8. Pour fournir votre propre document d'erreur personnalisé pour les erreurs de classe 4XX, entrez le nom du fichier du document d'erreur personnalisé dans Document d'erreur.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom de fichier du document d'erreur HTML que vous prévoyez de charger dans votre compartiment S3. Si vous ne spécifiez pas de document d'erreur personnalisé et qu'une erreur se produit, Amazon S3 renvoie un document d'erreur HTML par défaut. Pour plus d'informations, consultez Configuration d'un document d'erreur personnalisé.

9. (Facultatif) Si vous souhaitez spécifier des règles de redirection avancées, décrivez les règles à l'aide du langage JSON dans Redirection rules (Règles de redirection).

   Par exemple, vous pouvez acheminer les demandes de façon conditionnelle en fonction des noms ou préfixes de clés d'objets dans la demande. Pour plus d'informations, consultez Configurer des règles de redirection pour utiliser des redirections conditionnelles avancées.

10. Choisissez Enregistrer les modifications.

    Amazon S3 permet l'hébergement de site web statique pour votre compartiment. Au bas de la page, sous Static website hosting (Hébergement de site Web statique), vous voyez le point de terminaison du site web pour votre compartiment.

11. Sous Static website hosting (Hébergement de site Web statique), notez la valeur de Endpoint (Point de terminaison).

    Endpoint (Point de terminaison) correspond au point de terminaison du site web Amazon S3 de votre compartiment. Une fois que vous avez terminé de configurer votre compartiment en tant que site Web statique, vous pouvez utiliser ce point de terminaison pour tester votre site Web.

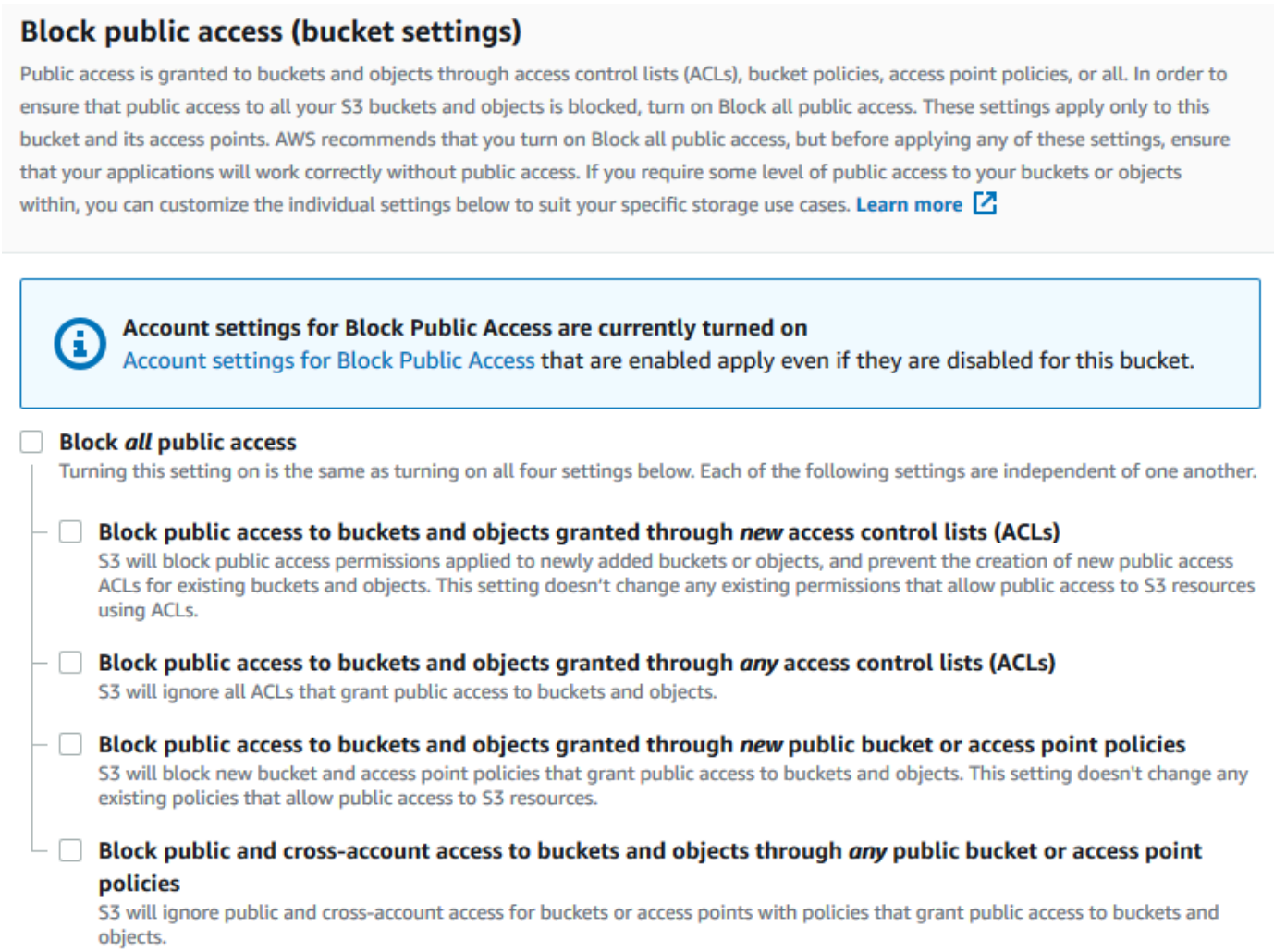
Étape 3 : Modifier les paramètres de blocage de l'accès public

Par défaut, Amazon S3 bloque l'accès public à votre compte et à vos compartiments. Si vous souhaitez utiliser un compartiment pour héberger un site web statique, vous pouvez utiliser ces étapes pour modifier vos paramètres de blocage de l'accès public.

> **Warning**
>
> Avant de terminer cette étape, revoyez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tout accès public à vos compartiments.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment que vous avez configuré en tant que site web statique.
3. Choisissez Permissions.
4. Sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)), choisissez Edit (Modifier).
5. Effacez Block all public access (Bloquer tous les accès publics) et choisissez Enregistrer les modifications.

> **Warning**
>
> Avant de terminer cette étape, examinez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tous les accès publics à vos compartiments.



Amazon S3 désactive les paramètres de blocage de l'accès public pour votre compartiment. Pour créer un site web public statique, vous devrez peut-être aussi modifier les paramètres de blocage de l'accès public de votre compte avant d'ajouter une stratégie de compartiment. Si les paramètres du compte pour la fonctionnalité de blocage de l'accès public sont actuellement activés, une note s'affiche sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)).

Étape 4 : Ajouter une stratégie de compartiment visant à rendre disponible publiquement le contenu de votre compartiment

Après avoir modifié les paramètres de blocage de l'accès public S3, vous devez ajouter une stratégie de compartiment pour accorder un accès public en lecture à votre compartiment. Lorsque vous accordez un accès public en lecture, tout le monde sur Internet peut accéder à votre compartiment.

> **Important**
>
> La stratégie suivante est uniquement un exemple et autorise un accès complet au contenu de votre compartiment. Avant d'effectuer cette étape, veuillez consulter Comment assurer la sécurité des fichiers de mon compartiment Amazon S3 ?, pour vous assurer que vous comprenez les bonnes pratiques pour sécuriser les fichiers dans votre compartiment S3 et les risques liés à l'octroi d'un accès public.

1. Dans Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Permissions.
3. Sous Bucket Policy (Stratégie de compartiment), choisissez Edit (Modifier).
4. Pour accorder l'accès public en lecture à votre site web, copiez la stratégie de compartiment suivante et collez-la dans l'Éditeur de stratégie de compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Mettez à jour `Resource` pour inclure le nom de votre compartiment.

   Dans l'exemple précédent de stratégie de compartiment, *`Bucket-Name`* est un espace réservé pour le nom du compartiment. Pour utiliser cette stratégie de compartiment avec votre propre compartiment, vous devez mettre à jour ce nom pour qu'il corresponde à celui de votre compartiment.

6. Choisissez Enregistrer les modifications.

   Un message s'affiche indiquant que la stratégie de compartiment a été ajoutée avec succès.

   Si une erreur indique `Policy has invalid resource`, confirmez que le nom du compartiment dans la stratégie de compartiment correspond au nom de votre compartiment. Pour plus d'informations sur l'ajout d'une politique de compartiment, consultez Comment ajouter une politique de compartiment S3 ?

   Si vous recevez un message d'erreur et que vous ne pouvez pas enregistrer la stratégie de compartiment, vérifiez les paramètres de blocage de l'accès public de votre compte et de votre compartiment pour confirmer que vous autorisez l'accès public au compartiment.

Étape 5 : Configurer un document d'index

Lorsque vous activez l'hébergement de site web statique pour votre compartiment, vous saisissez le nom du document d'index (par exemple, **`index.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous téléchargez un fichier HTML avec le nom du document de cet index dans votre compartiment.

Pour configurer le document d'index

1. Créez un fichier `index.html`.

   Si vous n'avez pas de ficher `index.html`, vous pouvez utiliser le code HTML suivant pour en créer un :

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
```

```
</body>
</html>
```

2. Enregistrez le fichier d'index au niveau local.

   Le nom du fichier du document d'index doit correspondre exactement au nom du document d'index que vous saisissez dans la boîte de dialogue Hébergement de site Web statique . Le nom du document d'index est sensible à la casse. Par exemple, si vous saisissez `index.html` pour le nom du Document d'index dans la boîte de dialogue Hébergement de site Web statique, le nom du fichier de votre document d'index doit également être `index.html` et non `Index.html`.

3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.

5. Activez l'hébergement de site Web statique pour votre compartiment et saisissez le nom exact de votre document d'index (par exemple, `index.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.

6. Pour charger le document d'index dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier d'index dans la liste du compartiment de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

7. (Facultatif) Chargez du contenus d'un autre site Web dans votre compartiment.

Étape 6 : Configurer un document d'erreur

Lorsque vous activez l'hébergement de site Web statique pour votre compartiment, vous entrez le nom du document d'erreur (par exemple, **`404.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous chargez un fichier HTML avec le nom du document d'erreur dans votre compartiment.

Pour configurer un document d'erreur

1. Créez un document d'erreur, par exemple `404.html`.
2. Enregistrez le fichier de document d'erreur au niveau local.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom que vous saisissez lorsque vous activez l'hébergement de site web statique. Par exemple, si vous entrez `404.html` pour le nom du document d'Erreur dans la boîte de dialogue Hébergement de site Web statique, le nom de fichier de votre document d'erreur doit également être `404.html`.
3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.
5. Activez l'hébergement de site Web statique pour votre compartiment et entrez le nom exact de votre document d'erreur (par exemple, `404.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web et Configuration d'un document d'erreur personnalisé.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.
6. Pour charger le document d'erreur dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier du document d'erreur dans la liste des compartiments de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

Étape 7 : Tester le point de terminaison de votre site web

Après avoir configuré l'hébergement de site web statique pour votre compartiment, vous pouvez tester le point de terminaison de votre site Web.

> **Note**
>
> Amazon S3 ne prend pas en charge l'accès HTTPS au site web. Si vous souhaitez utiliser le protocole HTTPS, vous pouvez utiliser Amazon CloudFront pour diffuser un site Web statique hébergé sur Amazon S3.
> Pour plus d'informations, consultez Comment utiliser CloudFront pour diffuser un site Web statique hébergé sur Amazon S3 ? et Exiger le protocole HTTPS pour la communication entre les spectateurs et CloudFront.

1. Dans Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Propriétés.
3. Au bas de la page, sous Héberger un site Web statique, choisissez le Point de terminaison du site Web du compartiment.

   Le document d'index s'ouvre dans une autre fenêtre du navigateur.

Vous avez désormais un site web hébergé sur Amazon S3. Ce site web est accessible au niveau du point de terminaison du site web Amazon S3. Néanmoins, vous devez avoir un domaine, de type `example.com`, pour proposer le contenu à partir du site web que vous avez créé. Vous pouvez également vouloir utiliser le domaine racine Amazon S3 pour traiter les demandes concernant `http://www.example.com` et `http://example.com`. Cela nécessite des étapes supplémentaires. Pour voir un exemple, consultez Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

Étape 8 : Nettoyage

Si vous avez créé votre site web statique comme un simple exercice d'apprentissage, supprimez les ressources AWS que vous avez allouées afin de ne plus accumuler de frais. Une fois que vous avez supprimé vos AWS ressources, votre site Web n'est plus disponible. Pour plus d’informations, consultez Suppression d'un compartiment.

Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53

Admettons que vous souhaitiez héberger un site web statique sur Amazon S3. Vous avez enregistré un domaine auprès d'Amazon Route 53 (par exemple, `example.com`) et vous souhaitez que les demandes concernant `http://www.example.com` et `http://example.com` soient servies à partir de votre contenu Amazon S3. Vous pouvez utiliser cette démonstration pour apprendre à héberger un site web statique et créer des redirections sur Amazon S3 pour un site web avec un nom de domaine personnalisé enregistré auprès de Route 53. Vous pouvez utiliser un site web existant que vous souhaitez héberger sur Amazon S3 ou utiliser cette démonstration pour commencer à partir de zéro.

Une fois cette procédure pas à pas terminée, vous pouvez éventuellement utiliser Amazon CloudFront pour améliorer les performances de votre site Web. Pour plus d'informations, consultez Accélérez votre site Web avec Amazon CloudFront.

> **Note**
>
> Les points de terminaison de site web Amazon S3 ne prennent pas en charge le protocole HTTPS ou les points d'accès. Si vous souhaitez utiliser le protocole HTTPS, vous pouvez utiliser Amazon CloudFront pour diffuser un site Web statique hébergé sur Amazon S3. Pour un didacticiel expliquant comment héberger votre contenu en toute sécurité avec CloudFront Amazon S3, consultezTutoriel : Hébergement de vidéos en streaming à la demande avec Amazon S3 CloudFront, Amazon et Amazon Route 53. Pour plus d'informations, consultez Comment utiliser CloudFront pour diffuser un site Web statique hébergé sur Amazon S3 ? et Exiger le protocole HTTPS pour la communication entre les spectateurs et CloudFront.

Automatiser la configuration d'un site Web statique à l'aide d'un modèle AWS CloudFormation

Vous pouvez utiliser un AWS CloudFormation modèle pour automatiser la configuration statique de votre site Web. Le AWS CloudFormation modèle définit les composants dont vous avez besoin pour héberger un site Web statique sécurisé afin que vous puissiez vous concentrer davantage sur le contenu de votre site Web et moins sur la configuration des composants.

Le AWS CloudFormation modèle inclut les composants suivants :

- Amazon S3 – Crée un compartiment Amazon S3 pour héberger votre site web statique.
- CloudFront — Crée une CloudFront distribution pour accélérer votre site Web statique.
- Lambda@Edge – Utilise Lambda@Edge pour ajouter des en-têtes de sécurité à chaque réponse du serveur. Les en-têtes de sécurité sont un groupe d'en-têtes dans la réponse du serveur Web qui indiquent aux navigateurs web d'appliquer des mesures de sécurité supplémentaires. Pour plus d'informations, consultez le billet de blog Ajouter des en-têtes de sécurité HTTP à l'aide de Lambda @Edge et Amazon. CloudFront

Ce AWS CloudFormation modèle est disponible pour que vous puissiez le télécharger et l'utiliser. Pour obtenir des informations et des instructions, consultez Getting started with a secure static website in the Amazon CloudFront Developer Guide.

Rubriques

Avant de commencer

À mesure que vous suivez les étapes de cet exemple, vous utilisez les services suivants :

Amazon Route 53 – Vous utilisez Route 53 pour enregistrer des domaines et définir où vous souhaitez acheminer le trafic Internet pour votre domaine. Cet exemple montre comment créer des enregistrements d'alias Route 53 qui acheminent le trafic de votre domaine (`example.com`) et de votre sous-domaine (`www.example.com`) vers un compartiment Amazon S3 qui contient un fichier HTML.

Amazon S3 – Vous utilisez Amazon S3 pour créer des compartiments, charger un modèle de page de site web, configurer des autorisations afin que toute personne puisse consulter le contenu et ensuite configurer les compartiments pour l'hébergement de sites web.

Étape 1 : Enregistrer un domaine personnalisé avec Route 53

Si vous ne disposez pas déjà d'un nom de domaine enregistré, tel que `example.com`, enregistrez-en un avec Route 53. Pour plus d'informations, consultez Enregistrement d'un nouveau domaine dans le Guide du développeur Amazon Route 53. Après avoir enregistré votre nom de domaine, vous pouvez créer et configurer vos compartiments Amazon S3 pour l'hébergement de sites web.

Étape 2 : Créer deux compartiments

Pour prendre en charge les demandes à partir du domaine racine et du sous-domaine, vous créez deux compartiments.

- Compartiment de domaine – `example.com`
- Compartiment de sous-domaine – `www.example.com`

Ces noms de compartiment doivent correspondre exactement à votre nom de domaine. Dans cet exemple, le nom de domaine est `example.com`. Vous hébergez votre contenu hors du compartiment de domaine racine (`example.com`). Vous créez une demande de redirection pour le compartiment de sous-domaine (`www.example.com`). Si quelqu'un saisit `www.example.com` dans son navigateur, il est redirigé vers `example.com` et voit le contenu hébergé dans le compartiment Amazon S3 portant ce nom.

Pour créer des compartiments pour l'hébergement de site web

Les instructions ci-dessous fournissent une vue d'ensemble de la façon de créer vos compartiments pour l'hébergement de site web. Pour obtenir des step-by-step instructions détaillées sur la création d'un bucket, consultezCréer un compartiment.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Créez votre compartiment de domaine racine.

   a. Choisissez Créer un compartiment.

   b. Saisissez le Nom du compartiment (par exemple **`example.com`**).

   c. Choisissez la Région dans laquelle vous souhaitez créer le compartiment.

      Choisissez une Région proche de vous sur le plan géographique afin de limiter la latence et les coûts, ou de répondre aux exigences réglementaires. C'est la Région que vous choisissez qui détermine votre point de terminaison de site web Amazon S3. Pour plus d'informations, consultez Points de terminaison de sites web.

   d. Pour accepter les paramètres par défaut et créer le compartiment, choisissez Créer.
3. Créez votre compartiment de sous-domaine :

   a. Choisissez Créer un compartiment.

   b. Saisissez le Nom du compartiment (par exemple **`www.example.com`**).

   c. Choisissez la Région dans laquelle vous souhaitez créer le compartiment.

      Choisissez une Région proche de vous sur le plan géographique afin de limiter la latence et les coûts, ou de répondre aux exigences réglementaires. C'est la Région que vous choisissez qui détermine votre point de terminaison de site web Amazon S3. Pour plus d'informations, consultez Points de terminaison de sites web.

   d. Pour accepter les paramètres par défaut et créer le compartiment, choisissez Créer.

Dans l'étape suivante, vous configurez `example.com` pour l'hébergement de site web.

Étape 3 : Configurer votre compartiment de domaine racine pour l'hébergement de site web

Dans cette étape, vous configurez votre compartiment de domaine racine (`example.com`) en tant que site web. Ce compartiment contient le contenu de votre site web. Lorsque vous configurez un compartiment pour héberger un site web, vous pouvez accéder à ce site à l'aide des Points de terminaison de sites web.

Pour activer l'hébergement de site Web statique

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer l'hébergement de sites web statiques.
3. Choisissez Propriétés.
4. Sous Static website hosting (Hébergement de site Web statique), choisissez Edit (Modifier).
5. Choisissez Utiliser ce compartiment pour héberger un site Web.
6. Sous Static website hosting (Hébergement de site web statique), choisissez Enable (Activer).
7. Dans Index document (Document d'index), entrez le nom du document d'index, généralement `index.html`.

   Le nom du document d'index est sensible à la casse et doit correspondre exactement au nom de fichier du document d'index HTML que vous prévoyez de charger dans votre compartiment S3. Lorsque vous configurez un compartiment pour l'hébergement d'un site web, vous devez indiquer un document d'index. Amazon S3 renvoie ce document d'index lorsque des demandes sont faites dans le domaine racine ou dans n'importe quel sous-dossier. Pour plus d'informations, consultez Configuration d'un document d'index.
8. Pour fournir votre propre document d'erreur personnalisé pour les erreurs de classe 4XX, entrez le nom du fichier du document d'erreur personnalisé dans Document d'erreur.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom de fichier du document d'erreur HTML que vous prévoyez de charger dans votre compartiment S3. Si vous ne spécifiez pas de document d'erreur personnalisé et qu'une erreur se produit, Amazon S3 renvoie un document d'erreur HTML par défaut. Pour plus d'informations, consultez Configuration d'un document d'erreur personnalisé.
9. (Facultatif) Si vous souhaitez spécifier des règles de redirection avancées, décrivez les règles à l'aide du langage JSON dans Redirection rules (Règles de redirection).

   Par exemple, vous pouvez acheminer les demandes de façon conditionnelle en fonction des noms ou préfixes de clés d'objets dans la demande. Pour plus d'informations, consultez Configurer des règles de redirection pour utiliser des redirections conditionnelles avancées.
10. Choisissez Enregistrer les modifications.

Amazon S3 permet l'hébergement de site web statique pour votre compartiment. Au bas de la page, sous Static website hosting (Hébergement de site Web statique), vous voyez le point de terminaison du site web pour votre compartiment.

11. Sous Static website hosting (Hébergement de site Web statique), notez la valeur de Endpoint (Point de terminaison).

    Endpoint (Point de terminaison) correspond au point de terminaison du site web Amazon S3 de votre compartiment. Une fois que vous avez terminé de configurer votre compartiment en tant que site Web statique, vous pouvez utiliser ce point de terminaison pour tester votre site Web.

Après avoir modifié les paramètres de blocage de l'accès public et ajouté une stratégie de compartiment qui autorise l'accès en lecture publique, vous pouvez utiliser le point de terminaison du site web pour accéder à votre site web.

Au cours de l'étape suivante, vous configurez votre sous-domaine (`www.example.com`) pour rediriger les demandes vers votre domaine (`example.com`).

Étape 4 : Configurer votre compartiment de sous-domaine pour la redirection de site web

Maintenant que vous avez configuré le compartiment de votre domaine racine pour l'hébergement de site Web, vous pouvez configurer le compartiment de votre sous-domaine pour rediriger toutes les demandes vers le domaine. Dans cet exemple, toutes les demandes pour `www.example.com` sont redirigées vers `example.com`.

Pour configurer une demande de redirection

1. Dans la console Amazon S3, dans la liste Buckets (Compartiments), choisissez nom de compartiment de votre sous-domaine (`www.example.com`, dans cet exemple).
2. Choisissez Propriétés.
3. Sous Static website hosting (Hébergement de site web statique), choisissez Edit (Modifier).
4. Choisissez Redirect requests for an object (Rediriger les demandes pour un objet).
5. Dans la zone Target bucket (Compartiment cible), entrez votre domaine racine, par exemple, **`example.com`**.
6. Pour Protocol (Protocole), choisissez http.
7. Choisissez Enregistrer les modifications.

Étape 5 : Configurer la journalisation pour le trafic du site web

Si vous souhaitez effectuer le suivi du nombre de visiteurs accédant à votre site web, vous pouvez activer la journalisation pour le compartiment de votre domaine racine. Pour plus d'informations, consultez Enregistrement de demandes avec journalisation des accès au serveur. Si vous envisagez d'utiliser Amazon CloudFront pour accélérer votre site Web, vous pouvez également utiliser la CloudFront journalisation.

Pour activer la journalisation des accès au serveur pour votre compartiment de domaine racine

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans la Région où vous avez créé le compartiment configuré en tant que site Web statique, créez un compartiment pour la journalisation, par exemple `logs.example.com`.
3. Créez un dossier pour les fichiers de journalisation des accès au serveur (par exemple, `logs`).
4. (Facultatif) Si vous souhaitez l'utiliser CloudFront pour améliorer les performances de votre site Web, créez un dossier pour les fichiers CloudFront journaux (par exemple,`cdn`).

   > **Important**
   >
   > Lorsque vous créez ou mettez à jour une distribution et que vous activez la CloudFront journalisation, la liste de contrôle d'accès au compartiment (ACL) est mise à CloudFront jour pour autoriser le `awslogsdelivery` compte `FULL_CONTROL` à écrire des journaux dans votre compartiment. Pour plus d'informations, consultez la section Autorisations requises pour configurer la journalisation standard et pour accéder à vos fichiers journaux dans le manuel Amazon CloudFront Developer Guide. Si le compartiment qui stocke les journaux utilise le paramètre imposé par le propriétaire du compartiment pour S3 Object Ownership afin de désactiver les ACL, il CloudFront ne peut pas écrire de journaux dans le compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.
5. Dans la liste Buckets (Compartiments) choisissez votre compartiment de domaine racine.
6. Choisissez Propriétés.
7. Sous Server access logging (Journalisation des accès au serveur), choisissez Edit (Modifier).
8. Sélectionnez Activer.
9. Sous Target bucket (Compartiment cible), choisissez la destination du compartiment et du dossier pour les journaux d'accès au serveur :

- Accédez à l'emplacement du dossier et du compartiment :
  1. Choisissez Browse S3 (Parcourir S3).
  2. Choisissez le nom du compartiment, puis le dossier des journaux.
  3. Choisissez Choose path (Sélectionnez un chemin).
- Saisissez le chemin du compartiment S3, par ex., `s3://logs.example.com/logs/`.

10. Choisissez Enregistrer les modifications.

    Dans votre compartiment de journaux, vous pouvez désormais accéder à vos journaux. Amazon S3 copie les journaux d'accès du site web dans votre compartiment de journaux toutes les deux heures.

Étape 6 : Charger l'index et le contenu du site web

Dans cette étape, vous chargez votre document d'index et le contenu de site web facultatif dans votre compartiment de domaine racine.

Lorsque vous activez l'hébergement de site web statique pour votre compartiment, vous saisissez le nom du document d'index (par exemple, **`index.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous téléchargez un fichier HTML avec le nom du document de cet index dans votre compartiment.

Pour configurer le document d'index

1. Créez un fichier `index.html`.

   Si vous n'avez pas de ficher `index.html`, vous pouvez utiliser le code HTML suivant pour en créer un :

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Enregistrez le fichier d'index au niveau local.

   Le nom du fichier du document d'index doit correspondre exactement au nom du document d'index que vous saisissez dans la boîte de dialogue Hébergement de site Web statique . Le nom du document d'index est sensible à la casse. Par exemple, si vous saisissez `index.html` pour le nom du Document d'index dans la boîte de dialogue Hébergement de site Web statique, le nom du fichier de votre document d'index doit également être `index.html` et non `Index.html`.

3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.

5. Activez l'hébergement de site Web statique pour votre compartiment et saisissez le nom exact de votre document d'index (par exemple, `index.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.

6. Pour charger le document d'index dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier d'index dans la liste du compartiment de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

7. (Facultatif) Chargez du contenus d'un autre site Web dans votre compartiment.

Étape 7 : Charger un document d'erreur

Lorsque vous activez l'hébergement de site Web statique pour votre compartiment, vous entrez le nom du document d'erreur (par exemple, **`404.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous chargez un fichier HTML avec le nom du document d'erreur dans votre compartiment.

Pour configurer un document d'erreur

1. Créez un document d'erreur, par exemple `404.html`.
2. Enregistrez le fichier de document d'erreur au niveau local.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom que vous saisissez lorsque vous activez l'hébergement de site web statique. Par exemple, si vous entrez `404.html` pour le nom du document d'Erreur dans la boîte de dialogue Hébergement de site Web statique, le nom de fichier de votre document d'erreur doit également être `404.html`.
3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.
5. Activez l'hébergement de site Web statique pour votre compartiment et entrez le nom exact de votre document d'erreur (par exemple, `404.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web et Configuration d'un document d'erreur personnalisé.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.
6. Pour charger le document d'erreur dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier du document d'erreur dans la liste des compartiments de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

Étape 8 : Modifier les paramètres de blocage de l'accès public S3

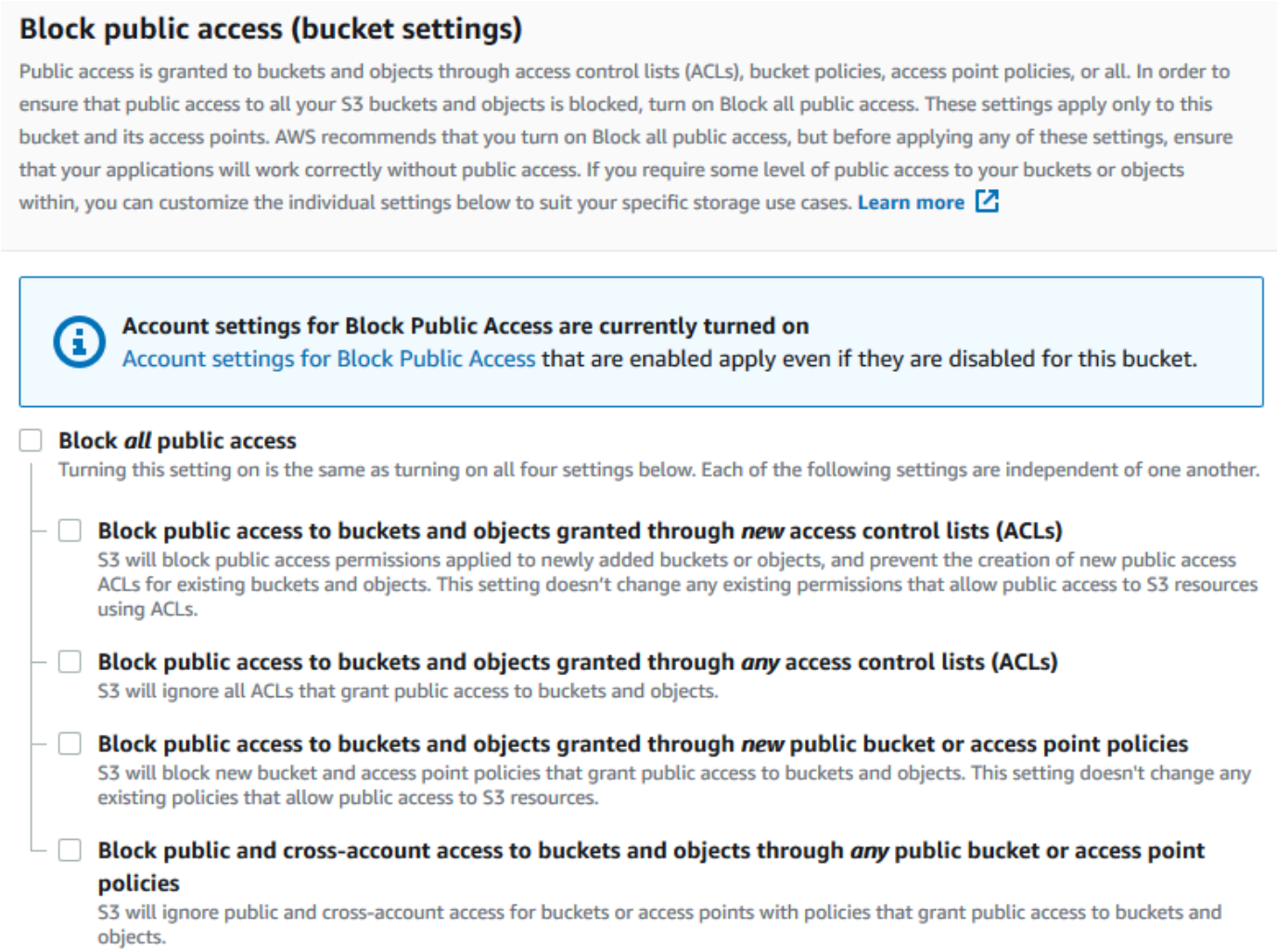
Dans cet exemple, vous modifiez les paramètres de blocage de l'accès public pour le compartiment de domaine (`example.com`) pour autoriser l'accès public.

Par défaut, Amazon S3 bloque l'accès public à votre compte et à vos compartiments. Si vous souhaitez utiliser un compartiment pour héberger un site web statique, vous pouvez utiliser ces étapes pour modifier vos paramètres de blocage de l'accès public.

> **Warning**
>
> Avant de terminer cette étape, revoyez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tout accès public à vos compartiments.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment que vous avez configuré en tant que site web statique.
3. Choisissez Permissions.
4. Sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)), choisissez Edit (Modifier).
5. Effacez Block all public access (Bloquer tous les accès publics) et choisissez Enregistrer les modifications.

> **Warning**
>
> Avant de terminer cette étape, examinez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tous les accès publics à vos compartiments.



Amazon S3 désactive les paramètres de blocage de l'accès public pour votre compartiment. Pour créer un site web public statique, vous devrez peut-être aussi modifier les paramètres de blocage de l'accès public de votre compte avant d'ajouter une stratégie de compartiment. Si les paramètres du compte pour la fonctionnalité de blocage de l'accès public sont actuellement activés, une note s'affiche sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)).

Étape 9 : Attacher une stratégie de compartiment

Dans cet exemple, vous attachez une stratégie de compartiment au compartiment de domaine (`example.com`) pour autoriser l'accès public en lecture. Vous remplacez la valeur *`Bucket-Name`*

dans l'exemple de stratégie de compartiment par le nom de votre compartiment de domaine, par exemple `example.com`.

Après avoir modifié les paramètres de blocage de l'accès public S3, vous devez ajouter une stratégie de compartiment pour accorder un accès public en lecture à votre compartiment. Lorsque vous accordez un accès public en lecture, tout le monde sur Internet peut accéder à votre compartiment.

> **Important**
>
> La stratégie suivante est uniquement un exemple et autorise un accès complet au contenu de votre compartiment. Avant d'effectuer cette étape, veuillez consulter Comment assurer la sécurité des fichiers de mon compartiment Amazon S3 ?, pour vous assurer que vous comprenez les bonnes pratiques pour sécuriser les fichiers dans votre compartiment S3 et les risques liés à l'octroi d'un accès public.

1. Dans Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Permissions.
3. Sous Bucket Policy (Stratégie de compartiment), choisissez Edit (Modifier).
4. Pour accorder l'accès public en lecture à votre site web, copiez la stratégie de compartiment suivante et collez-la dans l'Éditeur de stratégie de compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Mettez à jour `Resource` pour inclure le nom de votre compartiment.

Dans l'exemple précédent de stratégie de compartiment, *Bucket-Name* est un espace réservé pour le nom du compartiment. Pour utiliser cette stratégie de compartiment avec votre propre compartiment, vous devez mettre à jour ce nom pour qu'il corresponde à celui de votre compartiment.

6. Choisissez Enregistrer les modifications.

   Un message s'affiche indiquant que la stratégie de compartiment a été ajoutée avec succès.

   Si une erreur indique `Policy has invalid resource`, confirmez que le nom du compartiment dans la stratégie de compartiment correspond au nom de votre compartiment. Pour plus d'informations sur l'ajout d'une politique de compartiment, consultez Comment ajouter une politique de compartiment S3 ?

   Si vous recevez un message d'erreur et que vous ne pouvez pas enregistrer la stratégie de compartiment, vérifiez les paramètres de blocage de l'accès public de votre compte et de votre compartiment pour confirmer que vous autorisez l'accès public au compartiment.

Au cours de l'étape suivante, vous pouvez déterminer vos points de terminaison de site web et tester le point de terminaison de votre domaine.

Étape 10 : Tester le point de terminaison de domaine

Après avoir configuré votre compartiment de domaine pour héberger un site web public, vous pouvez tester votre point de terminaison. Pour plus d'informations, consultez Points de terminaison de sites web. Vous pourrez uniquement tester le point de terminaison pour votre compartiment de domaine puisque votre compartiment de sous-domaine est configuré pour rediriger le site web et non pour l'hébergement de site web statique.

> **Note**
>
> Amazon S3 ne prend pas en charge l'accès HTTPS au site web. Si vous souhaitez utiliser le protocole HTTPS, vous pouvez utiliser Amazon CloudFront pour diffuser un site Web statique hébergé sur Amazon S3.
> Pour plus d'informations, consultez Comment utiliser CloudFront pour diffuser un site Web statique hébergé sur Amazon S3 ? et Exiger le protocole HTTPS pour la communication entre les spectateurs et CloudFront.

1. Dans Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Propriétés.
3. Au bas de la page, sous Héberger un site Web statique, choisissez le Point de terminaison du site Web du compartiment.

   Le document d'index s'ouvre dans une autre fenêtre du navigateur.

Dans l'étape suivante, vous utilisez Amazon Route 53 pour permettre aux clients d'utiliser vos deux URL personnalisées pour naviguer vers le site.

Étape 11 : Ajouter des enregistrements d'alias pour vos domaine et sous-domaine

Dans cette étape, vous créez les enregistrements d'alias que vous ajoutez à la zone hébergée pour que le domaine mappe `example.com` et `www.example.com`. Au lieu d'utiliser des adresses IP, les enregistrements d'alias utilisent les points de terminaison du site Web Amazon S3. Amazon Route 53 conserve un mappage entre les enregistrements d'alias et les adresses IP où résident les compartiments Amazon S3. Vous créez deux enregistrements d'alias, l'un pour votre domaine racine et l'autre pour votre sous-domaine.

Ajouter un enregistrement d'alias pour votre domaine racine et votre sous-domaine

Pour ajouter un enregistrement d'alias pour votre domaine racine (**`example.com`**)

1. Ouvrez la console Route 53 sur https://console.aws.amazon.com/route53/home.

   > **Note**
   >
   > Si vous n'utilisez pas déjà Route 53, consultez Étape 1 : Enregistrer un domaine dans le guide du développeur Amazon Route 53. Une fois la configuration terminée, vous pouvez reprendre les instructions.

2. Choisissez Hosted Zones (Zones hébergées).
3. Dans la liste des zones hébergées, choisissez le nom de la zone hébergée qui correspond au nom de votre domaine.
4. Choisissez Create Record (Créer un enregistrement).
5. Choisissez Switch to wizard (Passer à l'assistant).

> **Note**
>
> Si vous souhaitez utiliser la création rapide pour créer vos enregistrements d'alias, consultez Configuration de Route 53 pour acheminer le trafic vers un compartiment S3.

6. Choisissez Simple routing (Routage simple), puis Next (Suivant).
7. Choisissez Define simple record (Définir un enregistrement simple).
8. Dans Record name (Nom de registre), acceptez la valeur par défaut, à savoir le nom de votre zone hébergée et de votre domaine.
9. Dans Value/Route traffic to (Valeur/Acheminer le trafic vers), choisissez Alias to S3 website endpoint (Alias vers le point de terminaison du site web S3).
10. Choisissez la Région .
11. Choisissez le compartiment S3.

    Le nom du compartiment doit correspondre au nom qui apparaît dans la zone Name (Nom) . Dans la liste Choose S3 bucket (Choisir un compartiment S3), le nom du compartiment apparaît avec le point de terminaison de site Web Amazon S3 pour la Région où le compartiment a été créé, par exemple `s3-website-us-west-1.amazonaws.com (example.com)`.

    Choose S3 bucket (Choisir un compartiment S3) répertorie un compartiment si :

    - Vous avez configuré le compartiment en tant que site web statique.
    - Le nom du compartiment est identique au nom du registre que vous créez.
    - Le courant Compte AWS a créé le compartiment.

    Si votre compartiment n'apparaît pas dans la liste Choose S3 bucket (Choisir un compartiment S3), saisissez le point de terminaison de site Web Amazon S3 de la Région dans laquelle le compartiment a été créé, par exemple, **`s3-website-us-west-2.amazonaws.com`**. Pour obtenir la liste complète des points de terminaison de sites Web Amazon S3, consultez Points de terminaison de sites Web Amazon S3. Pour en savoir plus sur la cible d'alias, consultez Valeur/acheminer le trafic vers dans le guide du développeur Amazon Route 53.
12. Dans Type d'enregistrement, choisissez A - Route le trafic vers une adresse IPv4 et certaines AWS ressources.
13. Pour Évaluer l'état de la cible, choisissez Non.
14. Choisissez Define simple record (Définir un enregistrement simple).

Pour ajouter un enregistrement d'alias pour votre sous-domaine (**`www.example.com`**)

1. Sous Configure records (Configurer les enregistrements), choisissez Define simple record (Définir un enregistrement simple).
2. Dans Record name (Nom de registre) pour votre sous-domaine, tapez `www`.
3. Dans Value/Route traffic to (Valeur/Acheminer le trafic vers), choisissez Alias to S3 website endpoint (Alias vers le point de terminaison du site web S3).
4. Choisissez la Région .
5. Choisissez le compartiment S3, par exemple, `s3-website-us-west-2.amazonaws.com (www.example.com)`.

   Si votre compartiment n'apparaît pas dans la liste Choose S3 bucket (Choisir un compartiment S3), saisissez le point de terminaison de site Web Amazon S3 de la Région dans laquelle le compartiment a été créé, par exemple, **`s3-website-us-west-2.amazonaws.com`**. Pour obtenir la liste complète des points de terminaison de sites Web Amazon S3, consultez Points de terminaison de sites Web Amazon S3. Pour en savoir plus sur la cible d'alias, consultez Valeur/acheminer le trafic vers dans le guide du développeur Amazon Route 53.
6. Dans Type d'enregistrement, choisissez A - Route le trafic vers une adresse IPv4 et certaines AWS ressources.
7. Pour Évaluer l'état de la cible, choisissez Non.
8. Choisissez Define simple record (Définir un registre simple).
9. Dans la page Configure records (Configurer des enregistrements), choisissez Create records (Créer des enregistrements).

> **Note**
>
> Les changements se propagent généralement sur tous les serveurs Route 53 en 60 secondes. Lorsque la propagation est terminée, vous pouvez acheminer le trafic vers votre compartiment Amazon S3 en utilisant les noms des enregistrements d'alias que vous avez créés au cours de cette procédure.

Ajouter un enregistrement d'alias pour votre domaine racine et votre sous-domaine (ancienne console Route 53)

Pour ajouter un enregistrement d'alias pour votre domaine racine (**`example.com`**)

La console Route 53 a été repensée. Dans la console Route 53, vous pouvez temporairement utiliser l'ancienne console. Si vous choisissez d'utiliser l'ancienne console Route 53, suivez la procédure ci-dessous.

1. Ouvrez la console Route 53 sur https://console.aws.amazon.com/route53/home.

   > **Note**
   >
   > Si vous n'utilisez pas déjà Route 53, consultez Étape 1 : Enregistrer un domaine dans le guide du développeur Amazon Route 53. Une fois la configuration terminée, vous pouvez reprendre les instructions.

2. Choisissez Hosted Zones (Zones hébergées).
3. Dans la liste des zones hébergées, choisissez le nom de la zone hébergée qui correspond au nom de votre domaine.
4. Choisissez Create Record Set (Créer un ensemble d'enregistrements).
5. Indiquez l'une des valeurs suivantes :

   Nom

   Acceptez la valeur par défaut, à savoir le nom de votre zone hébergée et de votre domaine.

   Pour le domaine racine, vous n'avez pas besoin d'entrer des informations supplémentaires dans le champ Name (Nom).

   Type

   Choisissez A – adresse IPv4.

   Alias

   Choisissez Yes (Oui).

   Cible d'alias

   Dans la section S3 website endpoints (Points de terminaison de site web S3) de la liste, choisissez le nom de votre compartiment.

Le nom du compartiment doit correspondre au nom qui apparaît dans la zone Name (Nom) . Dans la liste Alias Target (Cible d'alias), le nom du compartiment est suivi du point de terminaison du site web Amazon S3 pour la Région dans laquelle le compartiment a été créé, par exemple `example.com (s3-website-us-west-2.amazonaws.com)`. Alias Target (Cible d'alias) répertorie un compartiment si :

- Vous avez configuré le compartiment en tant que site web statique.
- Le nom du compartiment est identique au nom du registre que vous créez.
- Le courant Compte AWS a créé le compartiment.

Si votre compartiment n'apparaît pas dans la liste Alias Target (Cible d'alias), saisissez le point de terminaison du site web Amazon S3 pour la Région dans laquelle le compartiment a été créé, par exemple, `s3-website-us-west-2`. Pour obtenir la liste complète des points de terminaison de sites Web Amazon S3, consultez Points de terminaison de sites Web Amazon S3. Pour en savoir plus sur la cible d'alias, consultez Valeur/acheminer le trafic vers dans le guide du développeur Amazon Route 53.

Stratégie de routage

Acceptez la valeur par défaut Simple.

Évaluer l'état de la cible

Acceptez la valeur par défaut Non.

6. Choisissez Créer.

Pour ajouter un enregistrement d'alias pour votre sous-domaine (**`www.example.com`**)

1. Dans la zone hébergée pour votre domaine racine (`example.com`), choisissez Create Record Set (Créer un jeu d'enregistrements).
2. Indiquez l'une des valeurs suivantes :

   Nom

   Pour le sous-domaine, entrez `www` dans la zone.

   Type

   Choisissez A – adresse IPv4.

Alias

Choisissez Yes (Oui).

Cible d'alias

Dans la section S3 website endpoints (Points de terminaison de site web S3) de la liste, choisissez le même nom de compartiment que celui qui apparaît dans le champ Name (Nom), par exemple `www.example.com (s3-website-us-west-2.amazonaws.com)`.

Stratégie de routage

Acceptez la valeur par défaut Simple.

Évaluer l'état de la cible

Acceptez la valeur par défaut Non.

3. Choisissez Créer.

> **Note**
>
> Les changements se propagent généralement sur tous les serveurs Route 53 en 60 secondes. Lorsque la propagation est terminée, vous pouvez acheminer le trafic vers votre compartiment Amazon S3 en utilisant les noms des enregistrements d'alias que vous avez créés au cours de cette procédure.

Étape 12 : Tester le site web

Vérifiez que le site web et la redirection fonctionnent correctement. Dans votre navigateur, entrez vos URL. Dans cet exemple, vous testez les URL suivantes :

- Domaine (`http://example.com`) – Affiche le document d'index dans le compartiment `example.com`.
- Sous-domaine (`http://www.example.com`) – Redirige votre demande vers `http://example.com`. Vous voyez le document d'index dans le compartiment `example.com`.

Si votre site web ou les liens de redirection ne fonctionnent pas, vous pouvez essayer la procédure suivante :

- Effacer le cache – Effacez le cache de votre navigateur web.
- Vérifier les serveurs de noms – Si votre page web et les liens de redirection ne fonctionnent pas une fois que vous avez effacé votre cache, vous pouvez comparer les serveurs de noms de votre domaine et les serveurs de noms de votre zone hébergée. Si les serveurs de noms ne correspondent pas, vous devrez peut-être mettre à jour vos serveurs de noms de domaine pour qu'ils correspondent à ceux répertoriés dans votre zone hébergée. Pour plus d'informations, consultez Ajout ou modification de serveurs de noms et d'enregistrements de type glue pour un domaine.

Après avoir testé avec succès votre domaine racine et votre sous-domaine, vous pouvez configurer une CloudFront distribution Amazon pour améliorer les performances de votre site Web et fournir des journaux que vous pouvez utiliser pour examiner le trafic du site Web. Pour plus d'informations, consultez Accélérez votre site Web avec Amazon CloudFront.

Accélérez votre site Web avec Amazon CloudFront

Vous pouvez utiliser Amazon CloudFront pour améliorer les performances de votre site Web Amazon S3. CloudFront met les fichiers de votre site Web (tels que le HTML, les images et les vidéos) à disposition depuis les centres de données du monde entier (appelés emplacements périphériques). Lorsqu'un visiteur demande un fichier sur votre site Web, il redirige CloudFront automatiquement la demande vers une copie du fichier située à l'emplacement périphérique le plus proche. Le temps de téléchargement est alors plus rapide que si le visiteur avait demandé le contenu à un centre de données plus éloigné.

CloudFront met en cache le contenu aux emplacements périphériques pendant une période que vous spécifiez. Si un visiteur demande un contenu mis en cache depuis plus longtemps que la date d'expiration, CloudFront vérifie sur le serveur d'origine si une version plus récente du contenu est disponible. Si une version plus récente est disponible, CloudFront copie la nouvelle version vers l'emplacement périphérique. Les modifications que vous apportez au contenu d'origine sont répliquées aux emplacements périphériques lorsque les visiteurs demandent ce contenu.

Utilisation CloudFront sans Route 53

Le didacticiel de cette page utilise Route 53 pour pointer vers votre CloudFront distribution. Toutefois, si vous souhaitez diffuser du contenu hébergé dans un compartiment Amazon S3 CloudFront sans utiliser Route 53, consultez les CloudFrontdidacticiels Amazon : Configuration d'une distribution dynamique de contenu pour Amazon S3. Lorsque vous diffusez du contenu hébergé dans un

compartiment Amazon S3 à l'aide de CloudFront, vous pouvez utiliser n'importe quel nom de compartiment, et les protocoles HTTP et HTTPS sont pris en charge.

Automatiser la configuration à l'aide d'un modèle AWS CloudFormation

Pour plus d'informations sur l'utilisation d'un AWS CloudFormation modèle pour configurer un site Web statique sécurisé qui crée une CloudFront distribution destinée à votre site Web, consultez Getting started with a secure static website dans le manuel Amazon CloudFront Developer Guide.

Rubriques

Étape 1 : créer une CloudFront distribution

Tout d'abord, vous créez une CloudFront distribution. Votre site web est ainsi disponible à partir de centres de données dans le monde entier.

Pour créer une distribution avec une origine Amazon S3

1. Ouvrez la CloudFront console à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Choisissez Create Distribution.
3. Sur la page Créer une distribution dans la section Paramètres d'origine pour le Nom du domaine d'origine, saisissez le point de terminaison du site web Amazon S3 pour votre compartiment, par exemple, **`example.com.s3-website.us-west-1.amazonaws.com`**.

   CloudFront renseigne l'ID d'origine pour vous.
4. Pour Paramètres de comportement du cache par défaut, conservez les valeurs par défaut.

   Avec les paramètres par défaut de Stratégie de protocole d'utilisateur, vous pouvez utiliser HTTPS pour votre site web statique. Pour plus d'informations sur ces options de configuration, consultez la section Valeurs que vous spécifiez lorsque vous créez ou mettez à jour une distribution Web dans le manuel Amazon CloudFront Developer Guide.
5. Pour Paramètres de distribution, procédez comme suit :

   a. Laissez le paramètre Catégorie de tarifs défini sur Utiliser tous les emplacements périphériques (meilleure performance).

b. Définissez Noms de domaines alternatifs (CNAME) sur le domaine racine et le sous-domaine www. Dans le présent didacticiel, ce sont `example.com` et `www.example.com`.

> **Important**
>
> Avant d'effectuer cette étape, notez les exigences relatives à l'utilisation de noms de domaines alternatifs, en particulier la nécessité d'avoir un certificat SSL/TLS valide.

c. Pour Certificat SSL, choisissez Certificat SSL personnalisé (example.com), puis choisissez le certificat personnalisé qui couvre les noms de domaine et de sous-domaine.

   Pour plus d'informations, consultez la section Certificat SSL dans le manuel Amazon CloudFront Developer Guide.

d. Dans Objet racine par défaut, entrez le nom de votre document d'index, par exemple `index.html`.

   Si l'URL utilisée pour accéder à la distribution ne contient pas de nom de fichier, la CloudFront distribution renvoie le document d'index. L'objet racine par défaut doit correspondre exactement au nom du document d'index de votre site web statique. Pour plus d'informations, consultez Configuration d'un document d'index.

e. Définissez Journalisation sur Activé.

> **Important**
>
> Lorsque vous créez ou mettez à jour une distribution et que vous activez la CloudFront journalisation, la liste de contrôle d'accès au compartiment (ACL) est mise à CloudFront jour pour autoriser le `awslogsdelivery` compte `FULL_CONTROL` à écrire des journaux dans votre compartiment. Pour plus d'informations, consultez la section Autorisations requises pour configurer la journalisation standard et pour accéder à vos fichiers journaux dans le manuel Amazon CloudFront Developer Guide. Si le compartiment qui stocke les journaux utilise le paramètre imposé par le propriétaire du compartiment pour S3 Object Ownership afin de désactiver les ACL, il CloudFront ne peut pas écrire de journaux dans le compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

f. Pour Compartiment pour les journaux, choisissez le compartiment de journalisation que vous avez créé.

   Pour plus d'informations sur la configuration d'un compartiment de journalisation, consultez (Facultatif) Journalisation du trafic web.

g. Si vous souhaitez stocker les journaux générés par le trafic vers la CloudFront distribution dans un dossier, dans Log Prefix, entrez le nom du dossier.

h. Conservez tous les autres paramètres sur leurs valeurs par défaut.

6. Choisissez Create Distribution.

7. Pour voir le statut de la distribution, recherchez cette dernière dans la console et examinez la colonne Statut.

   Le statut `InProgress` indique que la distribution n'est pas encore pleinement déployée.

   Une fois votre distribution déployée, vous pouvez référencer votre contenu avec le nouveau nom de CloudFront domaine.

8. Enregistrez la valeur du nom de domaine affiché dans la CloudFront console, par exemple`dj4p1rv6mvubz.cloudfront.net`.

9. Pour vérifier que votre CloudFront distribution fonctionne, entrez le nom de domaine de la distribution dans un navigateur Web.

   Si votre site Web est visible, la CloudFront distribution fonctionne. Si votre site Web possède un domaine personnalisé enregistré auprès d'Amazon Route 53, vous aurez besoin du nom de CloudFront domaine pour mettre à jour l'ensemble d'enregistrements à l'étape suivante.

Étape 2 : Mettre à jour les jeux d'enregistrements pour votre domaine et votre sous-domaine

Maintenant que vous avez créé une CloudFront distribution avec succès, mettez à jour l'enregistrement d'alias dans Route 53 pour qu'il pointe vers la nouvelle CloudFront distribution.

Pour mettre à jour l'enregistrement d'alias afin qu'il pointe vers une CloudFront distribution

1. Ouvrez la console Route 53 sur https://console.aws.amazon.com/route53/home.

2. Dans le volet gauche de navigation, choisissez Hosted zones (Zones hébergées).

3. Sur la page Hosted Zones (Zones hébergées), choisissez la zone hébergée que vous avez créée pour votre sous-domaine, par exemple `www.example.com`.
4. Sous Records (Enregistrements), sélectionnez l'enregistrement A que vous avez créé pour votre sous-domaine.
5. Sous Record details (Détails de l'enregistrement), choisissez Edit record (Modifier l'enregistrement).
6. Sous Acheminer le trafic vers, choisissez Alias vers CloudFront la distribution.
7. Sous Choisir une distribution, choisissez la CloudFront distribution.
8. Choisissez Enregistrer.
9. Pour rediriger l'enregistrement A du domaine racine vers la CloudFront distribution, répétez cette procédure pour le domaine racine, par exemple`example.com`.

   La mise à jour des jeux d'enregistrements prend effet dans un délai de 2 à 48 heures.
10. Pour voir si les nouveaux enregistrements A ont pris effet, entrez l'URL de votre sous-domaine dans un navigateur web, par exemple `http://www.example.com`.

    Si le navigateur ne vous redirige plus vers le domaine racine (par exemple `http://example.com`), les nouveaux enregistrements A sont en place. Lorsque le nouvel enregistrement A prend effet, le trafic acheminé par le nouvel enregistrement A vers la CloudFront distribution n'est pas redirigé vers le domaine racine. Tous les visiteurs qui font référence au site en utilisant `http://example.com` ou `http://www.example.com` sont redirigés vers l'emplacement CloudFront périphérique le plus proche, où ils bénéficient de temps de téléchargement plus rapides.

> **Tip**
>
> Les navigateurs peuvent mettre en cache les paramètres de redirection. Si vous pensez que les paramètres du nouvel enregistrement A devraient avoir pris effet mais que votre navigateur redirige encore `http://www.example.com` vers `http://example.com`, essayez d'effacer le cache et l'historique de votre navigateur, de fermer et de rouvrir votre application de navigateur, ou d'utiliser un navigateur web différent.

(Facultatif) Étape 3 : Vérifier les fichiers journaux

Les journaux d'accès vous indiquent combien de personnes visitent le site web. Ils contiennent également des données professionnelles utiles que vous pouvez analyser avec d'autres services, tels que Amazon EMR.

CloudFront les journaux sont stockés dans le compartiment et le dossier que vous choisissez lorsque vous créez une CloudFront distribution et que vous activez la journalisation. CloudFront écrit des journaux dans votre bucket de journaux dans les 24 heures suivant l'envoi des demandes correspondantes.

Pour consulter les fichiers journaux de votre site Web

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment de journaux pour votre site web.
3. Choisissez le dossier CloudFront des journaux.
4. Téléchargez les `.gzip` fichiers écrits par CloudFront avant de les ouvrir.

   Si vous avez créé votre site web comme un simple exercice d'apprentissage, vous pouvez supprimer les ressources que vous avez allouées afin de ne plus accumuler de frais. Pour ce faire, consultez Nettoyage de vos exemples de ressources. Une fois que vous avez supprimé vos ressources AWS , votre site web n'est plus disponible.

Nettoyage de vos exemples de ressources

Si vous avez créé votre site web statique dans le cadre d'un exercice d'apprentissage, veillez à supprimer les ressources AWS que vous avez allouées afin de ne plus accumuler de frais. Une fois que vous avez supprimé vos ressources AWS , votre site web n'est plus disponible.

Tâches

- Étape 1 : supprimer la CloudFront distribution Amazon
- Étape 2 : Supprimer la zone hébergée Route 53
- Étape 3 : Désactivez la journalisation et supprimez votre compartiment S3

Étape 1 : supprimer la CloudFront distribution Amazon

Avant de supprimer une CloudFront distribution Amazon, vous devez la désactiver. Une distribution désactivée n'est plus fonctionnelle et n'accumule pas de frais. Vous pouvez activer une distribution désactivée à tout moment. Une fois que vous avez supprimé une distribution désactivée, celle-ci n'est plus disponible.

Pour désactiver et supprimer une CloudFront distribution

1. Ouvrez la CloudFront console à l'adressehttps://console.aws.amazon.com/cloudfront/v4/home.
2. Sélectionnez la distribution que vous souhaitez désactiver, puis choisissez Désactiver).
3. Lorsque vous serez invité à confirmer l'opération, choisissez Oui, désactiver.
4. Sélectionnez la distribution désactivée, puis choisissez Supprimer.
5. Lorsque vous êtes invité à confirmer l'opération, choisissez Oui, supprimer.

Étape 2 : Supprimer la zone hébergée Route 53

Avant de supprimer la zone hébergée, vous devez supprimer les jeux d'enregistrements que vous avez créés. Vous n'avez pas besoin de supprimer les enregistrements NS et SOA ; ils sont automatiquement supprimés lorsque vous supprimez la zone hébergée.

Pour supprimer les jeux d'enregistrements

1. Ouvrez la console Route 53 sur https://console.aws.amazon.com/route53/home.
2. Dans la liste des noms de domaines, sélectionnez votre nom de domaine, puis choisissez Go to Record Sets (Accéder aux jeux d'enregistrements).
3. Dans la liste des jeux d'enregistrement, sélectionnez les enregistrements A que vous avez créés.

   Le type de chaque jeu d'enregistrements est répertorié dans la colonne Type.
4. Choisissez Delete Record Set.
5. Lorsque vous êtes invité à confirmer l'opération, choisissez Confirm (Confirmer).

Pour supprimer une zone hébergée Route 53

1. Suite à la procédure précédente, choisissez Back to Hosted Zones (Retour aux zones hébergées).
2. Sélectionnez votre nom de domaine, puis choisissez Supprimer une zone hébergée.

3. Lorsque vous êtes invité à confirmer l'opération, choisissez Confirm (Confirmer).

Étape 3 : Désactivez la journalisation et supprimez votre compartiment S3

Avant de supprimer votre compartiment S3, assurez-vous que la journalisation est désactivée pour le compartiment. Dans le cas contraire, AWS continue d'écrire des journaux dans votre compartiment lorsque vous le supprimez.

Pour désactiver la journalisation pour un compartiment

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Sous Compartiments, choisissez le nom de votre compartiment, puis choisissez Propriétés.
3. Dans Properties (Propriétés), choisissez Logging (Journalisation).
4. Désactivez la case à cocher Activé.
5. Choisissez Enregistrer.

Vous pouvez maintenant supprimer votre compartiment. Pour plus d’informations, consultez Suppression d'un compartiment.

Création, configuration et utilisation des compartiments Amazon S3

Pour stocker vos données dans Amazon S3, vous devez utiliser avec des ressources appelées compartiments et objets. Un compartiment est un conteneur d'objets. Un objet est un fichier et toutes les métadonnées qui le décrivent.

Pour stocker un objet dans Amazon S3, vous devez créer un compartiment, puis télécharger l'objet dans un compartiment. Lorsque l'objet se trouve dans le compartiment, vous pouvez l'ouvrir, le télécharger et le déplacer. Lorsque vous n'avez plus besoin d'un objet ou d'un compartiment, vous pouvez nettoyer vos ressources.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

> **Note**
>
> Avec Amazon S3, vous ne payez que les services que vous utilisez. Pour plus d'informations sur les fonctionnalités et les tarifs d'Amazon S3, consultez Amazon S3. Si vous êtes un nouveau client Amazon S3, vous pouvez commencer à utiliser Amazon S3 gratuitement. Pour plus d'informations, consultez la page sur l'offer gratuite AWS.

Les rubriques de cette section fournissent une vue d'ensemble de l'utilisation des compartiments dans Amazon S3. Elles incluent des informations sur l'attribution de noms, la création, l'accès et la suppression de compartiments. Pour plus d'informations sur l'affichage ou l'établissement de la liste des objets d'un compartiment, consultez Organisation, liste et utilisation de vos objets.

Rubriques

- Créer un compartiment
- Affichage des propriétés d'un compartiment S3
- Vider un compartiment
- Suppression d'un compartiment
- Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3
- Utilisation de Mountpoint pour Amazon S3
- Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration
- Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation
- Limites et restrictions applicables aux compartiments

Présentation des compartiments

Pour charger vos données (photos, vidéos, documents, etc.) dans Amazon S3, vous devez d'abord créer un compartiment S3 dans l'une des Régions AWS.

Un compartiment est un conteneur d'objets stockés dans Amazon S3. Chaque compartiment permet de stocker un nombre illimité d'objets et vous pouvez avoir jusqu'à 100 compartiments dans votre compte. Pour demander une augmentation, reportez-vous à la console Service Quotas.

Chaque objet est contenu dans un compartiment. Par exemple, si l'objet nommé `photos/puppy.jpg` est stocké dans le compartiment `DOC-EXAMPLE-BUCKET`, dans la région USA Ouest (Oregon), il est adressable à l'aide de l'URL `https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/photos/puppy.jpg`. Pour plus d'informations, consultez Accès à un compartiment.

En termes de mise en œuvre, les compartiments et les objets sont AWS des ressources, et Amazon S3 fournit des API pour vous permettre de les gérer. Par exemple, vous pouvez créer un compartiment et y charger des objets via l'API Amazon S3. Vous pouvez également effectuer ces opérations via la console Amazon S3. La console utilise les API Amazon S3 pour envoyer des demandes à Amazon S3.

Cette section décrit comment utiliser les compartiments. Pour en savoir plus sur l'utilisation des objets, consultez Présentation des objets Amazon S3.

Amazon S3 prend en charge les compartiments globaux, ce qui signifie que chaque nom de compartiment doit être unique Comptes AWS Régions AWS dans l'ensemble d'une partition. Une partition est un regroupement de Régions. AWS dispose actuellement de trois partitions : `aws` (Régions Standard), `aws-cn` (Régions Chine) et `aws-us-gov` (AWS GovCloud (US)).

Une fois qu'un bucket est créé, le nom de ce bucket ne peut pas être utilisé par un autre utilisateur Compte AWS de la même partition tant que le bucket n'est pas supprimé. Vous ne devez pas dépendre des conventions de dénomination de compartiment spécifiques à des fins de vérification de la disponibilité ou de la sécurité. Pour obtenir des instructions sur l'attribution des noms, consultez Règles de dénomination de compartiment.

Amazon S3 crée des compartiments dans une région que vous spécifiez. Pour réduire la latence, minimiser les coûts ou répondre aux exigences réglementaires, choisissez Région AWS celui qui est géographiquement proche de vous. Par exemple, si vous résidez en Europe, il peut être avantageux de créer des compartiments dans les régions Europe (Irlande) ou Europe (Francfort). Pour obtenir la liste des régions Amazon S3, veuillez consulter Régions et points de terminaison dans les Références générales AWS .

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

> **Note**
>
> Les objets appartenant à un compartiment que vous créez dans une région spécifique Région AWS ne quittent jamais cette région, sauf si vous les transférez explicitement vers une autre région. Par exemple, les objets stockés dans la région Europe (Irlande) ne la quittent jamais.

Rubriques

- À propos des autorisations
- Gestion de l'accès public aux compartiments
- Options de configuration des compartiments

À propos des autorisations

Vous pouvez utiliser vos Utilisateur racine d'un compte AWS informations d'identification pour créer un compartiment et effectuer toute autre opération Amazon S3. Toutefois, nous vous recommandons de ne pas utiliser les informations d'identification de l'utilisateur root Compte AWS pour effectuer des demandes, par exemple pour créer un bucket. Créez plutôt un utilisateur AWS Identity and Access Management (IAM) et accordez-lui un accès complet (les utilisateurs n'ont aucune autorisation par défaut).

Ces utilisateurs sont appelés administrateurs. Vous pouvez utiliser les informations d'identification de l'utilisateur administrateur, au lieu des informations d'identification de l'utilisateur root de votre compte, pour interagir avec AWS et effectuer des tâches, telles que créer un bucket, créer des utilisateurs et leur accorder des autorisations.

Pour plus d'informations, consultez la section sur les informations d'identification de l'Utilisateur racine d'un compte AWS et les informations d'identification d'un utilisateur IAM dans Références générales AWS , ainsi que Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Celui Compte AWS qui crée une ressource possède cette ressource. Par exemple, si vous créez un utilisateur IAM dans votre compartiment Compte AWS et que vous lui accordez l'autorisation de créer un compartiment, l'utilisateur peut créer un compartiment. Mais l'utilisateur n'est pas propriétaire du bucket ; le bucket auquel Compte AWS il appartient est propriétaire du bucket. Pour que l'utilisateur puisse effectuer d'autres opérations sur les compartiments, le propriétaire de la ressource doit lui accorder d'autres autorisations. Pour plus d'informations sur la gestion des autorisations relatives à vos ressources Amazon S3, consultez Identity and Access Management pour Amazon S3.

Gestion de l'accès public aux compartiments

Un accès public est accordé aux compartiments et objets via des politiques de compartiment, des listes de contrôle d'accès (ACL) ou les deux. Pour vous aider à gérer l'accès public aux ressources Amazon S3, Amazon S3 fournit des paramètres permettant de bloquer l'accès public. Les paramètres de blocage de l'accès public Amazon S3 peuvent remplacer les listes ACL et les stratégies de compartiment pour vous permettre d'appliquer des limites uniformes concernant l'accès public à ces ressources. Vous pouvez appliquer des paramètres de blocage de l'accès public à des compartiments individuels ou à tous les compartiments de votre compte.

Pour vous assurer que l'accès public à tous vos compartiments et objets Amazon S3 est bloqué, les quatre paramètres liés au blocage de l'accès public sont activés par défaut lorsque vous créez un nouveau compartiment. Nous vous recommandons d'activer ces quatre paramètres également pour

votre compte. Ces paramètres bloquent tout accès public pour tous les compartiments présents et futurs.

Avant d'appliquer ces paramètres, vérifiez que vos applications fonctionnent correctement sans accès public. Si vous avez besoin d'un certain niveau d'accès public à vos compartiments ou objets (par exemple pour héberger un site Web statique, comme décrit dans Hébergement d'un site Web statique à l'aide d'Amazon S3), vous pouvez personnaliser les paramètres individuels afin de les adapter à vos cas d'utilisation du stockage. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Toutefois, nous vous recommandons vivement de garder l'option Bloquer l'accès public activée. Si vous souhaitez conserver les quatre paramètres de blocage de l'accès public activés et héberger un site Web statique, vous pouvez utiliser le contrôle CloudFront d'accès d'origine (OAC) d'Amazon. Amazon CloudFront fournit les fonctionnalités requises pour configurer un site Web statique sécurisé. Les sites Web statiques Amazon S3 prennent uniquement en charge les points de terminaison HTTP. Amazon CloudFront utilise le stockage durable d'Amazon S3 tout en fournissant des en-têtes de sécurité supplémentaires, tels que HTTPS. HTTPS accroît la sécurité en chiffrant une demande HTTP normale et en offrant une protection contre les cyberattaques courantes.

Pour plus d'informations, consultez Getting started with a secure static website in the Amazon CloudFront Developer Guide.

> **Note**
>
> Si vous voyez une `Error` lorsque vous répertoriez vos compartiments et leurs paramètres d'accès public, il se peut que vous ne disposiez pas des autorisations requises. Assurez-vous d'avoir ajouté les autorisations suivantes à votre politique utilisateur ou à votre politique de rôle :
>
> ```
> s3:GetAccountPublicAccessBlock
> s3:GetBucketPublicAccessBlock
> s3:GetBucketPolicyStatus
> s3:GetBucketLocation
> s3:GetBucketAcl
> s3:ListAccessPoints
> s3:ListAllMyBuckets
> ```
>
> Dans de rares cas, les demandes peuvent également échouer en raison d'une défaillance de la Région AWS .

Options de configuration des compartiments

Amazon S3 vous permet de configurer votre compartiment grâce à différentes options. Par exemple, vous pouvez configurer votre compartiment pour l'hébergement d'un site Web, le configurer afin qu'il enregistre tous les accès au compartiment, ou encore ajouter une configuration pour la gestion du cycle de vie des objets qu'il contient. Amazon S3 prend en charge des sous-ressources qui vous permettent de stocker et de gérer les informations relatives à la configuration du compartiment. Vous pouvez utiliser l'API Amazon S3 pour créer et gérer ces sous-ressources. Cependant, vous pouvez également utiliser la console ou les AWS SDK.

> **Note**
>
> Les configurations au niveau de l'objet sont également possibles. Par exemple, vous pouvez configurer une liste de contrôle d'accès (ACL) spécifique à un objet, ce qui vous permet de configurer des autorisations au niveau de cet objet.

Nous parlons de « sous-ressources », car celles-ci se trouvent dans le contexte d'un compartiment ou d'un objet donnés. Le tableau ci-dessous répertorie les sous-ressources qui vous permettent de gérer les configurations associées à un compartiment donné.

Sous-ressource	Description
cors (partage des ressources cross-origin)	Vous pouvez configurer le compartiment afin d'autoriser les demandes cross-origin. Pour plus d'informations, consultez Utilisation du partage des ressources entre origines multiples (CORS).
event notification	Vous pouvez autoriser le compartiment à vous envoyer des notifications lorsque certains événements s'y produisent. Pour plus d'informations, consultez Notifications d'événements Amazon S3.
lifecycle	Vous pouvez définir des règles de cycle de vie pour des objets de votre compartiment, lorsque ceux-ci disposent d'un cycle de vie bien défini. Par exemple, vous pouvez définir une règle qui permet d'archiver certains objets un an après leur création ou de les supprimer 10 ans après leur création.

Sous-ressource	Description
	Pour plus d’informations, consultez Gestion du cycle de vie de votre stockage.
location	Lorsque vous créez un compartiment, vous spécifiez l' Région AWS endroit où vous souhaitez qu'Amazon S3 le crée. Amazon S3 stocke cette informati on dans la sous-ressource associée à l'emplacement et fournit une API qui vous permet de récupérer cette information.
logging	La journalisation vous permet d'effectuer le suivi des demandes d'accès au compartiment. Chaque enregistrement de journal d'accès fournit des informations détaillées sur une demande d'accès donnée (demandeur, nom du compartiment, heure de la demande, action associée à la demande, état de la réponse et code d'erreur, le cas échéant). Les informations des journaux d'accès peuvent s'avérer utiles en cas d'audit de sécurité ou d'audit des accès. Cela peut également vous aider à en savoir plus sur votre base de clients et à comprendre votre facture Amazon S3. Pour plus d’informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.
Verrouillage d’objet	Pour utiliser le verrouillage des objets S3, vous devez l'activer pour un compartiment. Si vous le souhaitez, vous pouvez également configurer un mode et une période de rétention par défaut qui s'appliqueront aux nouveaux objets mis en place dans le compartiment. Pour plus d’informations, consultez Utilisation du verrouillage des objets S3.
policy et ACL (liste de contrôle d'accès)	Par défaut, toutes vos ressources (telles que les objets et compartim ents) sont privées. Amazon S3 prend en charge les options de liste de contrôle d'accès (ACL) et de stratégie de compartiment, qui vous permettent d'accorder et de gérer des autorisations au niveau du compartiment. Amazon S3 stocke les informations relatives aux autorisations dans les sous-ress ources policy et acl. Pour plus d’informations, consultez Identity and Access Management pour Amazon S3.

Sous-ressource	Description
réplication	La réplication est la copie automatique et asynchrone d'objets entre des compartiments dans des Régions AWS différents ou identiques. Pour plus d'informations, consultez Vue d'ensemble de la réplication d'objets.
requestPayment	Par défaut, Compte AWS celui qui crée le bucket (le propriétaire du bucket) paie les téléchargements depuis le bucket. Toutefois, cette sous-ressource permet au propriétaire du compartiment de spécifier que l'utilisateur qui demande un téléchargement soit facturé lorsqu'il effectue un téléchargement. Amazon S3 fournit une API qui vous permet de gérer cette sous-ressource. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.
tagging	Vous pouvez ajouter des balises de répartition des coûts à votre compartim ent pour classer et suivre vos AWS coûts. Amazon S3 fournit la sous-ress ource tagging qui vous permet de stocker et de gérer des balises sur un compartiment. À l'aide des balises que vous appliquez à votre compartiment, vous AWS générez un rapport de répartition des coûts avec l'utilisation et les coûts agrégés par balises. Pour plus d'informations, consultez Rapports de facturation et d'utilisation pour Amazon S3.
Transfer Acceleration	Transfer Acceleration permet un transfert rapide, facile et sécurisé de fichiers à grande distance entre votre client et un compartiment S3. Transfer Acceleration tire parti des emplacements périphériques distribués dans le monde entier d'Amazon CloudFront. Pour plus d'informations, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Sous-ressource	Description
gestion des versions	La gestion des versions vous permet de récupérer un objet remplacé ou supprimé par erreur. La gestion des versions est une bonne pratique que nous recommandons pour la récupération d'objets remplacés ou supprimés par erreur. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.
website	Vous pouvez configurer votre compartiment pour l'hébergement de sites Web statiques. Amazon S3 crée une sous-ressource website afin de stocker cette configuration. Pour plus d'informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.

Règles de dénomination de compartiment

Les règles suivantes s'appliquent à la dénomination des compartiments à usage général et des compartiments de répertoires dans Amazon S3 :

Rubriques

- Règles de dénomination des compartiments à usage général
- Règles de dénomination des compartiments de répertoires

Règles de dénomination des compartiments à usage général

Les règles de dénomination suivantes s'appliquent aux compartiments à usage général.

- Les noms de compartiment peuvent comporter entre 3 (min.) et 63 (max.) caractères.
- Les noms de compartiment peuvent être composés uniquement de lettres minuscules, de chiffres, de points (.) et de traits d'union (-).
- Les noms de compartiment doivent commencer et se terminer par une lettre ou un chiffre.
- Les noms de compartiment ne doivent pas contenir deux points consécutifs.

- Les noms de compartiments ne doivent pas utiliser le même format que les adresses IP (par ex., 192.168.5.4).
- Les noms de compartiment ne doivent pas commencer par le préfixe `xn--`.
- Les noms des compartiments ne doivent pas commencer par le préfixe `sthree-` ou le `sthree-configurator` préfixe.
- Les noms de compartiment ne doivent pas se terminer par le suffixe `-s3alias`. Ce suffixe est réservé aux noms d'alias de point d'accès. Pour plus d'informations, consultez Utilisation d'un alias de type compartiment pour votre point d'accès de compartiment S3.
- Les noms de compartiment ne doivent pas se terminer par le suffixe `--ol-s3`. Ce suffixe est réservé aux noms d'alias de point d'accès Object Lambda. Pour plus d'informations, consultez Comment utiliser un alias de type compartiment pour votre point d'accès Object Lambda de compartiment S3.
- Les noms des compartiments doivent être uniques pour Comptes AWS l'ensemble Régions AWS de la partition. Une partition est un regroupement de régions. AWS possède actuellement trois partitions : `aws` (Régions standard), `aws-cn` (Régions chinoises) et `aws-us-gov` (AWS GovCloud (US)).
- Le nom d'un bucket ne peut pas être utilisé par un autre Compte AWS utilisateur de la même partition tant que le bucket n'est pas supprimé.
- Les compartiments utilisés avec Amazon S3 Transfer Acceleration ne peuvent pas avoir de points (.) dans leurs noms. Pour plus d'informations sur Transfer Acceleration, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Pour une meilleure compatibilité, nous vous recommandons d'éviter d'utiliser des points (.) dans les noms de compartiment, à l'exception des compartiments utilisés uniquement pour l'hébergement de sites web statiques. Si vous incluez des points dans le nom d'un bucket, vous ne pouvez pas utiliser l' virtual-host-style adressage via HTTPS, sauf si vous validez vous-même le certificat. En effet, les certificats de sécurité utilisés pour l'hébergement virtuel de compartiments ne sont pas compatibles avec les compartiments dont le nom contient un point.

Cette limitation n'affecte pas les compartiments utilisés pour l'hébergement de sites web statiques, car ce type d'hébergement n'est disponible que sur HTTP. Pour plus d'informations sur l' virtual-host-styleadressage, consultezHébergement virtuel de compartiments. Pour plus d'informations sur l'hébergement de sites web statiques, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.

> **Note**
>
> Avant le 1er mars 2018, les compartiments créés dans la région USA Est (Virginie du Nord) pouvaient comporter des noms incluant jusqu'à 255 caractères et comprenant des lettres majuscules et des traits de soulignement. À compter du 1er mars 2018, les nouveaux compartiments de la région USA Est (Virginie du Nord) doivent être conformes aux règles appliquées dans toutes les autres régions.

Pour en savoir plus sur les noms de clé d'objet, consultez la section Creating object key names (Création de noms de clés d'objet).

Exemples de noms de compartiments à usage général

Les exemples de noms de compartiment suivants sont valides et suivent les consignes de dénomination recommandées pour les compartiments à usage général :

- `docexamplebucket1`
- `log-delivery-march-2020`
- `my-hosted-content`

Les exemples de noms de compartiment ci-dessous sont valides, mais non recommandés pour un usage autre que l'hébergement de site web statique :

- `docexamplewebsite.com`
- `www.docexamplewebsite.com`
- `my.example.s3.bucket`

Les exemples de noms de compartiment suivants ne sont pas valides :

- `doc_example_bucket` (contient des traits de soulignement)
- `DocExampleBucket` (contient des lettres majuscules)
- `doc-example-bucket-` (se termine par un trait d'union)

Règles de dénomination des compartiments de répertoires

Le nom d'un compartiment de répertoires :

- Soyez unique au sein de la zone Région AWS de disponibilité choisie.
- Le nom doit comporter entre 3 (min) et 63 (max) caractères, suffixe compris.
- Doit être composé uniquement de lettres minuscules, de chiffres et de traits d'union (-).
- Commencer et se terminer par une lettre ou un chiffre.
- Doit inclure le suffixe suivant :`--azid--x-s3`.

> **Note**
>
> Lorsque vous créez un bucket de répertoire à l'aide de la console, un suffixe est automatiquement ajouté au nom de base que vous fournissez. Ce suffixe inclut l'ID de la zone de disponibilité que vous avez choisie.
> Lorsque vous créez un bucket d'annuaire à l'aide d'une API, vous devez fournir le suffixe complet, y compris l'ID de zone de disponibilité, dans votre demande. Pour obtenir la liste des ID de zone de disponibilité, consultezRégions et zones de disponibilité S3 Express One Zone.

Accès à un compartiment Amazon S3 et liste des compartiments

Pour répertorier et accéder à vos compartiments Amazon S3, vous pouvez utiliser différents outils. Passez en revue les outils suivants pour déterminer quelle approche correspond à votre cas d'utilisation :

- Console Amazon S3 : avec la console Amazon S3, vous pouvez facilement accéder à un compartiment et modifier ses propriétés. Vous pouvez également effectuer la plupart des opérations sur les compartiments à l'aide de l'interface utilisateur de la console, sans écrire de code.
- AWS CLI: Si vous devez accéder à plusieurs compartiments, vous pouvez gagner du temps en utilisant le AWS Command Line Interface (AWS CLI) pour automatiser les tâches courantes et répétitives. La scriptabilité et la reproductibilité des actions courantes sont fréquemment prises en compte à mesure que les entreprises évoluent. Pour plus d'informations, consultez Développement avec Amazon S3 à l'aide de la AWS CLI.

- API REST Amazon S3 : vous pouvez utiliser l'API REST Amazon S3 pour écrire vos propres programmes et accéder aux compartiments par programmation. Amazon S3 prend en charge une architecture d'API dans laquelle vos compartiments et vos objets sont des ressources, chacune d'entre elles disposant d'un URI de ressource permettant de l'identifier de façon unique. Pour plus d'informations, consultez Développer avec Amazon S3 à l'aide de l'API REST.

Selon le cas d'utilisation de votre compartiment Amazon S3, différentes méthodes sont recommandées pour accéder aux données sous-jacentes de vos compartiments. La liste suivante inclut les cas d'utilisation courants pour accéder à vos données.

- Sites web statiques : vous pouvez utiliser Amazon S3 pour héberger un site web statique. Dans ce cas d'utilisation, vous pouvez configurer votre compartiment S3 afin qu'il fonctionne comme un site web. Pour obtenir un exemple qui explique les différentes étapes d'hébergement d'un site web sur Amazon S3, consultez Didacticiel : configuration d'un site web statique sur Amazon S3.

  Pour héberger un site Web statique avec des paramètres de sécurité tels que le blocage de l'accès public activé, nous vous recommandons d'utiliser Amazon CloudFront avec Origin Access Control (OAC) et d'implémenter des en-têtes de sécurité supplémentaires, tels que HTTPS. Pour plus d'informations, consultez Démarrer avec un site web statique sécurisé.

  > **Note**
  >
  > Amazon S3 prend en charge les URL de type hébergement virtuel et les URL de type chemin. Les compartiments étant accessibles via des URL de type chemin et de type hébergement virtuel, nous vous recommandons de créer des compartiments avec des noms compatibles DNS. Pour plus d'informations, consultez Limites et restrictions applicables aux compartiments.

- Jeux de données partagés : au fur et à mesure de la mise à l'échelle sur Amazon S3, il est courant d'adopter un modèle à locataires multiples, dans lequel vous affectez à différents clients finaux ou unités commerciales des préfixes uniques au sein d'un compartiment partagé. En utilisant les points d'accès Amazon S3, vous pouvez diviser une politique de compartiment de grande taille en politiques de point d'accès distinctes et discrètes pour chaque application devant accéder au jeu de données partagé. Cette approche permet de se concentrer plus simplement sur l'élaboration de la stratégie d'accès appropriée à une application sans perturber les activités des autres applications dans le jeu de données partagé. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

- Charges de travail à débit élevé : Mountpoint pour Amazon S3 est un client de fichiers open source à débit élevé permettant de monter un compartiment Amazon S3 en tant que système de fichiers local. Avec Mountpoint, vos applications peuvent accéder aux objets stockés dans Amazon S3 par le biais d'opérations de système de fichiers, telles qu'ouvrir et lire. Mountpoint traduit automatiquement ces opérations en appels d'API d'objet S3, permettant à vos applications d'accéder au stockage et au débit élastiques d'Amazon S3 via une interface de fichier. Pour plus d'informations, consultez Utilisation de Mountpoint pour Amazon S3.
- Applications à plusieurs régions : les points d'accès multi-régions Amazon S3 fournissent un point de terminaison global que les applications peuvent utiliser pour traiter les demandes provenant de compartiments S3 situés dans plusieurs Régions AWS. Vous pouvez utiliser des points d'accès multi-régions pour créer des applications multi-régions avec la même architecture utilisée dans une seule région, puis exécuter ces applications partout dans le monde. Au lieu d'envoyer les demandes sur l'Internet public, les points d'accès multi-régions offrent une résilience de réseau intégrée avec une accélération des demandes Internet vers Amazon S3. Pour plus d'informations, consultez Points d'accès multi-régions dans Amazon S3.
- Création de nouvelles applications : vous pouvez utiliser les AWS SDK lorsque vous développez des applications avec Amazon S3. Les AWS SDK simplifient vos tâches de programmation en encapsulant l'API REST Amazon S3 sous-jacente. Pour créer des applications mobiles et Web connectées, vous pouvez utiliser les SDK AWS mobiles et la AWS Amplify JavaScript bibliothèque. Pour plus d'informations, consultez Développement avec Amazon S3 à l'aide des AWS SDK.
- Protocole de transfert de fichiers (SFTP) Secure Shell (SSH) : si vous essayez de transférer des données sensibles en toute sécurité sur Internet, vous pouvez utiliser un serveur compatible SFTP avec votre compartiment Amazon S3. AWS SFTP est un protocole réseau qui prend en charge toutes les fonctionnalités de sécurité et d'authentification du SSH. Avec ce protocole, vous avez un contrôle précis sur l'identité des utilisateurs, les autorisations et les clés ou vous pouvez utiliser des politiques IAM pour gérer l'accès. Pour associer un serveur compatible SFTP à votre compartiment Amazon S3, assurez-vous de créer votre serveur compatible SFTP en premier. Ensuite, vous configurez des comptes d'utilisateurs et vous associez le serveur à un compartiment Amazon S3. Pour une présentation détaillée de ce processus, consultez AWS Transfer for SFTP — Service SFTP entièrement géré pour Amazon S3 dans AWS les blogs.

Liste des compartiments

Pour répertorier tous vos compartiments, vous devez disposer de l'autorisation `s3:ListAllMyBuckets`. Pour accéder à un bucket, assurez-vous également d'obtenir les autorisations requises AWS Identity and Access Management (IAM) pour répertorier le contenu du

bucket spécifié. Pour obtenir un exemple de politique de compartiment qui accorde l'accès à un compartiment S3, consultez Autoriser un accès utilisateur IAM à l'un de vos compartiments. Si vous rencontrez l'erreur indiquant que l'accès HTTP est refusé (403 Forbidden), consultez Stratégies de compartiment et politiques IAM.

Vous pouvez répertorier votre compartiment à l'aide de la console Amazon S3 AWS CLI, du ou AWS des SDK.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste des compartiments à usage général, choisissez le compartiment que vous souhaitez consulter.

> **Note**
>
> La liste des compartiments à usage général inclut les compartiments qui se trouvent dans tous les compartiments. Régions AWS

À l'aide du AWS CLI

Pour accéder AWS CLI à un compartiment S3 ou générer une liste de compartiments S3, utilisez la `ls` commande. Lorsque vous répertoriez tous les objets de votre compartiment, notez que vous devez disposer de l'autorisation `s3:ListBucket`.

Pour utiliser cet exemple de commande, remplacez *`DOC-EXAMPLE-BUCKET1`* par le nom de votre compartiment.

```
$ aws s3 ls s3://DOC-EXAMPLE-BUCKET1
```

L'exemple de commande suivant répertorie tous les compartiments Amazon S3 de votre compte :

```
$ aws s3 ls
```

Pour plus d'informations et pour obtenir des exemples, consultez Liste des compartiments et des objets.

Utilisation des AWS SDK

Vous pouvez également accéder à un compartiment Amazon S3 à l'aide de l'opération d'API ListBuckets. Pour des exemples d'utilisation de cette opération avec différents AWS SDK, consultezUtilisation `ListBuckets` avec un AWS SDK ou une CLI.

Créer un compartiment

Pour charger vos données sur Amazon S3, vous devez d'abord créer un compartiment Amazon S3 dans l'une des Régions AWS. Lorsque vous créez un compartiment, vous devez choisir un nom de compartiment et une Région. Vous pouvez choisir facultativement d'autres options de gestion du stockage pour le compartiment. Une fois un compartiment créé, vous ne pouvez pas modifier son nom ni sa Région. Pour plus d'informations sur la dénomination des compartiments, consultez Règles de dénomination de compartiment.

Celui Compte AWS qui crée le bucket en est propriétaire. Vous pouvez charger un nombre illimité d'objets vers ce compartiment. Par défaut, vous pouvez créer jusqu'à 100 compartiments dans chacun de vos Comptes AWS. Si vous avez besoin de compartiments supplémentaires, vous pouvez augmenter votre limite de compartiments de compte à un maximum de 1 000 compartiments en soumettant une demande d'augmentation de limite de service. Pour découvrir comment envoyer une demande d'augmentation de limite de compartiment, consultez Quotas de Service AWS dans la Référence générale d'AWS . Chaque compartiment permet de stocker un nombre illimité d'objets.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes de contrôle d'accès (ACL). Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient chaque objet présent dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques.

Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) est le niveau de base de la configuration du chiffrement pour chaque compartiment dans Amazon S3. Tous les nouveaux objets chargés dans un compartiment S3 sont automatiquement chiffrés avec SSE-S3 comme paramètre de chiffrement de base. Si vous souhaitez utiliser un autre type de chiffrement par défaut, vous pouvez également spécifier le chiffrement côté serveur avec des clés AWS Key Management

Service (AWS KMS) (SSE-KMS) ou des clés fournies par le client (SSE-C) pour chiffrer vos données. Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Vous pouvez utiliser la console Amazon S3, les API Amazon S3 ou AWS les SDK pour créer un compartiment. AWS CLI Pour plus d'informations sur les autorisations requises pour créer un bucket, consultez le CreateBucketmanuel Amazon Simple Storage Service API Reference.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer un bucket.

   > **Note**
   >
   > Pour limiter la latence et les coûts, et répondre aux exigences légales, choisissez une région proche de vous. Les objets stockés dans une Région ne la quittent jamais, sauf si vous les transférez explicitement vers une autre Région. Pour obtenir la liste d'Amazon S3 Régions AWS, consultez la section sur les Service AWS points de terminaison dans le Référence générale d'Amazon Web Services.

3. Dans le panneau de navigation de gauche, choisissez Compartiments.
4. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
5. Sous Configuration générale, consultez l' Région AWS endroit où votre bucket sera créé.
6. Sous Type de compartiment, sélectionnez Usage général.
7. Pour Nom du compartiment, saisissez le nom de votre compartiment.

   Le nom du compartiment doit présenter les caractéristiques suivantes :

   - Être unique dans une partition. Une partition est un regroupement de Régions. AWS dispose actuellement de trois partitions : `aws` (Régions Standard), `aws-cn` (Régions Chine) et `aws-us-gov` (AWS GovCloud (US) Regions).
   - Il doit comporter entre 3 et 63 caractères.

- Être uniquement composé de lettres minuscules, de chiffres, de points (.) et de traits d'union (-). Pour une meilleure compatibilité, nous vous recommandons d'éviter d'utiliser des points (.) dans les noms de compartiment, à l'exception des compartiments utilisés uniquement pour l'hébergement de sites web statiques.
- Commencer et se terminer par une lettre ou un chiffre.

Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour plus d'informations sur l'attribution de noms à des compartiments, consultez Règles de dénomination de compartiment.

> **Important**
>
> Évitez d'inclure des informations sensibles, notamment des numéros de compte, dans le nom du compartiment. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

8. AWS Management Console vous permet de copier les paramètres d'un bucket existant dans votre nouveau bucket. Si vous ne souhaitez pas copier les paramètres d'un bucket existant, passez à l'étape suivante.

> **Note**
>
> Cette option :
>
> - N'est pas disponible dans le AWS CLI et n'est disponible que dans la console
> - Non disponible pour les compartiments de répertoire
> - Ne copie pas la politique du bucket du bucket existant vers le nouveau bucket

Pour copier les paramètres d'un compartiment existant, sous Copier les paramètres d'un compartiment existant, sélectionnez Choisir un compartiment. La fenêtre Choose bucket s'ouvre. Recherchez le compartiment contenant les paramètres que vous souhaitez copier, puis sélectionnez Choisir un compartiment. La fenêtre Choisir un compartiment se ferme et la fenêtre Créer un compartiment s'ouvre à nouveau.

Sous Copier les paramètres d'un bucket existant, vous pouvez maintenant voir le nom du bucket que vous avez sélectionné. Vous verrez également une option Restaurer les paramètres par

défaut que vous pouvez utiliser pour supprimer les paramètres du bucket copiés. Passez en revue les autres paramètres du compartiment sur la page Créer un compartiment. Vous verrez qu'ils correspondent désormais aux paramètres du bucket que vous avez sélectionné. Vous pouvez passer à la dernière étape.

9. Sous Object Ownership (Propriété de l'objet), pour désactiver ou activer les listes ACL et contrôler la propriété des objets téléchargés dans votre compartiment, sélectionnez l'un des paramètres suivants :

   Listes ACL désactivées

   - Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations d'accès aux données du compartiment S3. Le compartiment utilise des stratégies exclusivement pour définir le contrôle des accès.

     Par défaut, les listes ACL sont désactivées. La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

   Listes ACL activées

   - Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.

     Si vous appliquez le paramètre Propriétaire du compartiment préféré, pour exiger que tous les chargements Amazon S3 incluent la liste ACL prédéfinie `bucket-owner-full-control`, vous pouvez ajouter une politique de compartiment qui autorise uniquement les chargements d'objets utilisant cette liste ACL.

   - Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

> **Note**
>
> Le paramètre par défaut est Propriétaire du compartiment appliqué. Pour appliquer le paramètre par défaut et maintenir les listes ACL désactivées, seule l'autorisation `s3:CreateBucket` est requise. Pour activer les listes ACL, vous devez disposez de l'autorisation `s3:PutBucketOwnershipControls`.

10. Dans Paramètres de blocage de l'accès public pour ce compartiment, choisissez les paramètres Bloquer l'accès public que vous souhaitez appliquer au compartiment.

    Par défaut, les quatre paramètres de blocage de l'accès public sont activés. Nous vous recommandons de maintenir tous les paramètres activés, sauf si vous savez que vous devez en désactiver un ou plusieurs pour votre cas d'utilisation spécifique. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

    > **Note**
    >
    > Pour activer tous les paramètres de blocage de l'accès public, seule l'autorisation `s3:CreateBucket` est requise. Pour désactiver les paramètres de blocage de l'accès public, vous devez disposer de l'autorisation `s3:PutBucketPublicAccessBlock`.

11. (Facultatif) Sous Bucket Versioning (Gestion des versions du compartiment), vous pouvez choisir de conserver les variantes des objets dans votre compartiment. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3.

    Pour désactiver ou activer la gestion des versions sur votre compartiment, choisissez Disable (Désactiver) ou Enable (Activer).

12. (Facultatif) Sous Tags (Balises), vous pouvez choisir d'ajouter des balises à votre compartiment. Les balises sont des paires clé-valeur utilisées pour catégoriser le stockage.

    Pour ajouter une balise de compartiment, saisissez une Key (Clé) et éventuellement une Value (Valeur), puis choisissez Add Tag (Ajouter une balise).

13. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).

14. Pour configurer le chiffrement par défaut, dans Type de chiffrement, choisissez l'une des options suivantes :

- Clés gérées par Amazon S3 (SSE-S3)
- AWS Key Management Service clé (SSE-KMS)

> **Important**
>
> Si vous utilisez l'option SSE-KMS pour votre configuration de chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les AWS KMS quotas et sur la manière de demander une augmentation de quota, consultez la section Quotas dans le guide du AWS Key Management Service développeur.

Les compartiments et les nouveaux objets sont chiffrés à l'aide d'un chiffrement côté serveur avec une clé gérée par Amazon S3 comme niveau de base de configuration du chiffrement. Pour plus d'informations sur le chiffrement par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Pour en savoir plus sur l'utilisation du chiffrement côté serveur Amazon S3 pour chiffrer vos données, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

15. Si vous avez sélectionné CléAWS Key Management Service (SSE-KMS), procédez comme suit :

    a. Sous CléAWS KMS , spécifiez votre clé KMS de l'une des manières suivantes :

    - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

      La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .

    - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
    - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous ne pouvez utiliser que les clés KMS disponibles dans le même compartiment Région AWS que le bucket. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé KMS, puis saisir l'ARN de la clé KMS. Pour plus d'informations sur les autorisations entre comptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service . Pour en savoir plus sur SSE-KMS, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS).
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 prend uniquement en charge les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur l'utilisation AWS KMS avec Amazon S3, consultezUtilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

b. Lorsque vous configurez votre compartiment pour utiliser le chiffrement par défaut avec SSE-KMS, vous pouvez également activer les clés de compartiment S3. Les clés de compartiment S3 réduisent le coût du chiffrement en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

   Pour utiliser les clés de compartiment S3, sous la Clé de compartiment, choisissez Activer.

16. (Facultatif) Si vous souhaitez activer le verrouillage des objets S3, procédez comme suit :

    a. Choisissez Advanced Settings (Paramètres avancés).

    > **Important**
    >
    > L'activation du verrouillage d'objet active également la gestion des versions pour le compartiment. Après l'avoir activé, vous devez configurer les paramètres de conservation et de mise en suspens juridique par défaut du verrouillage d'objets pour protéger les nouveaux objets contre la suppression ou l'écrasement.

    b. Pour activer le verrouillage d'objets, choisissez Enable (Activer), lisez l'avertissement qui s'affiche et confirmez-le.

    Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

    > **Note**
    >
    > Pour créer un compartiment prenant en charge le verrouillage d'objets, vous devez disposer des autorisations suivantes : `s3:CreateBucket`, `s3:PutBucketVersioning` et `s3:PutBucketObjectLockConfiguration`.

17. Choisissez Créer un compartiment.

Utilisation des AWS SDK

Lorsque vous utilisez les AWS SDK pour créer un compartiment, vous devez créer un client, puis utiliser le client pour envoyer une demande de création d'un compartiment. En tant que bonne pratique, vous devez créer votre client et votre compartiment dans la même Région AWS. Si vous ne spécifiez pas de région lorsque vous créez un client ou un compartiment, Amazon S3 utilise USA Est (Virginie du Nord), la région par défaut. Si vous souhaitez limiter la création du compartiment à une Région AWS spécifique, utilisez la clé de condition `LocationConstraint`.

Pour créer un client afin d'accéder à un point de terminaison Dual-Stack, vous devez spécifier une Région AWS. Pour plus d'informations, consultez Points de terminaison Dual-Stack. Pour obtenir la liste des régions disponibles Régions AWS, consultez la section Régions et points de terminaison dans le Références générales AWS.

Lorsque vous créez un client, la Région est mappée au point de terminaison spécifique à la Région. Le client utilise ce point de terminaison pour communiquer avec Amazon S3 : `s3.region.amazonaws.com`. Si votre Région a été lancée après le 20 mars 2019, votre client et votre compartiment doivent se trouver dans la même Région. Par conséquent, vous pouvez utiliser un client dans la Région USA Est (Virginie du Nord) pour créer un compartiment dans n'importe quelle Région lancée avant le 20 mars 2019. Pour plus d'informations, consultez Points de terminaison hérités.

Ces exemples de code AWS SDK exécutent les tâches suivantes :

- Créer un client en indiquant explicitement une Région AWS : dans l'exemple, le client utilise le point de terminaison `s3.us-west-2.amazonaws.com` pour communiquer avec Amazon S3. Vous pouvez spécifier n'importe quelle Région AWS. Pour en obtenir la liste Régions AWS, voir Régions et points de terminaison dans le manuel de référence AWS général.
- Envoyer une demande de création de compartiment en indiquant uniquement un nom de compartiment : le client envoie une demande à Amazon S3 pour créer le compartiment dans la région où vous avez créé un client.
- Récupérer des informations relatives à l'emplacement du compartiment : Amazon S3 stocke les informations relatives à l'emplacement du compartiment dans la sous-ressource de l'emplacement associée au compartiment.

Java

Cet exemple montre comment créer un compartiment Amazon S3 à l'aide du kit AWS SDK for Java. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.GetBucketLocationRequest;

import java.io.IOException;
```

```
public class CreateBucket2 {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            if (!s3Client.doesBucketExistV2(bucketName)) {
                // Because the CreateBucketRequest object doesn't specify a region,
 the
                // bucket is created in the region specified in the client.
                s3Client.createBucket(new CreateBucketRequest(bucketName));

                // Verify that the bucket was created by retrieving it and checking
 its
                // location.
                String bucketLocation = s3Client.getBucketLocation(new
 GetBucketLocationRequest(bucketName));
                System.out.println("Bucket location: " + bucketLocation);
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Pour plus d'informations sur la création et le test d'un échantillon fonctionnel, consultez le manuel de référence de l'API AWS SDK for .NET Version 3.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CreateBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CreateBucketAsync().Wait();
        }

        static async Task CreateBucketAsync()
        {
            try
            {
                if (!(await AmazonS3Util.DoesS3BucketExistAsync(s3Client,
 bucketName)))
                {
                    var putBucketRequest = new PutBucketRequest
                    {
                        BucketName = bucketName,
                        UseClientRegion = true
                    };

                    PutBucketResponse putBucketResponse = await
 s3Client.PutBucketAsync(putBucketRequest);
                }
                // Retrieve the bucket location.
                string bucketLocation = await FindBucketLocationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
```

```
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
        static async Task<string> FindBucketLocationAsync(IAmazonS3 client)
        {
            string bucketLocation;
            var request = new GetBucketLocationRequest()
            {
                BucketName = bucketName
            };
            GetBucketLocationResponse response = await
 client.GetBucketLocationAsync(request);
            bucketLocation = response.Location.ToString();
            return bucketLocation;
        }
    }
}
```

Ruby

Pour plus d'informations sur la création et le test d'un échantillon fonctionnel, consultez AWS SDK for Ruby - Version 3.

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
 This is a client-side object until
  #                                 create is called.
  def initialize(bucket)
    @bucket = bucket
  end
```

```
  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      "None. You must create a bucket before you can get its location!"
    else
      @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("doc-example-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

À l'aide du AWS CLI

Vous pouvez également utiliser le AWS Command Line Interface (AWS CLI) pour créer un compartiment S3. Pour plus d'informations, consultez create-bucket dans la Référence des commandes AWS CLI .

Pour plus d'informations sur le AWS CLI, voir Qu'est-ce que le AWS Command Line Interface ? dans le guide de AWS Command Line Interface l'utilisateur.

Affichage des propriétés d'un compartiment S3

Vous pouvez consulter les propriétés de n'importe quel compartiment Amazon S3 que vous possédez. Il s'agit notamment des paramètres suivants :

- Bucket Versioning (Gestion des versions de compartiment) – Conservez plusieurs versions d'un objet dans un même compartiment à l'aide de la gestion des versions. Par défaut, la gestion des versions est désactivée pour un nouveau compartiment. Pour plus d'informations sur l'activation de la gestion des versions, consultez Activation de la gestion des versions sur les compartiments.
- Balises — Dans AWS le cadre de la répartition des coûts, vous pouvez utiliser des balises de compartiment pour annoter la facturation liée à l'utilisation d'un compartiment. Une balise correspond à une paire clé-valeur représentant un libellé que vous affectez à un compartiment. Pour plus d'informations, consultez Utilisation des balises de répartition des coûts pour les compartiments S3.
- Default encryption (Chiffrement par défaut) – L'activation du chiffrement par défaut vous permet de chiffrer automatiquement côté serveur. Amazon S3 chiffre un objet avant de l'enregistrer sur un disque et déchiffre l'objet quand vous le téléchargez. Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.
- Server access logging (Journalisation des accès au serveur) – Obtenez des enregistrements détaillés des demandes portant sur votre compartiment avec la journalisation des accès au serveur. Par défaut, Amazon S3 ne collecte pas les journaux d'accès au serveur. Pour plus d'informations sur l'activation de la journalisation des accès au serveur, consultez Activation de la journalisation des accès au serveur Amazon S3.
- AWS CloudTrail événements liés aux données : permet CloudTrail de consigner les événements liés aux données. Par défaut, les suivis ne consignent pas les événements de données. Des frais supplémentaires sont facturés pour les événements de données. Pour plus d'informations, consultez la section Consignation d'événements de données pour les journaux d'activité du Guide de l'utilisateur AWS CloudTrail .

- Event notifications (Notifications d'événements) – Autorisez certains événements de compartiment Amazon S3 à envoyer des messages de notification à une destination chaque fois que ces événements se produisent. Pour plus d'informations, consultez Activation et configuration des notifications d'événements à l'aide de la console Amazon S3.
- Transfer acceleration (Accélération de transfert) – Activez un transfert rapide, facile et sécurisé de fichiers à grande distance entre votre client et un compartiment S3. Pour plus d'informations sur l'activation de l'accélération du transfert, consultez Activation et utilisation de S3 Transfer Acceleration.
- Object Lock (Verrouillage des objets) – Utilisez le verrouillage des objets S3 pour empêcher qu'un objet soit supprimé ou remplacé sur une période déterminée ou indéfinie. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.
- Requester Pays (Paiement par le demandeur) – Activez cette option pour que le demandeur (plutôt que le propriétaire du compartiment) paie les demandes et les transferts de données. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.
- Static website hosting (Hébergement de site web statique) – Vous pouvez héberger un site web statique dans Amazon S3. Pour plus d'informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.

Vous pouvez afficher les propriétés du bucket à l'aide des AWS SDK AWS Management Console AWS CLI, ou

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment dont vous souhaitez afficher les propriétés.
3. Choisissez l'onglet Propriétés.
4. Sur la page Propriétés, vous pouvez configurer les propriétés ci-dessus pour le bucket.

À l'aide du AWS CLI

Consultez les propriétés du bucket à l'aide du AWS CLI

Les commandes suivantes montrent comment vous pouvez utiliser le AWS CLI pour répertorier les différentes propriétés du compartiment.

Ce qui suit renvoie le jeu de balises associé au bucket *example-s3-bucket1*. Pour plus d'informations sur les bucket tags, voir,Utilisation des balises de répartition des coûts pour les compartiments S3.

```
aws s3api get-bucket-tagging --bucket example-s3-bucket1
```

Pour plus d'informations et des exemples, consultez get-bucket-tagging dans la Référence des commandes AWS CLI .

Ce qui suit renvoie l'état de version du bucket *example-s3-bucket1*. Pour plus d'informations sur le versionnement des compartiments, consultezUtilisation de la gestion des versions dans les compartiments S3.

```
aws s3api get-bucket-versioning --bucket example-s3-bucket1
```

Pour plus d'informations et des exemples, consultez get-bucket-versioning dans la Référence des commandes AWS CLI .

Ce qui suit renvoie la configuration de chiffrement par défaut pour le bucket *example-s3-bucket1*. Par défaut, tous les compartiments ont une configuration de chiffrement par défaut qui utilise le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3). Pour plus d'informations sur le chiffrement par défaut du bucket, consultezDéfinition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

```
aws s3api get-bucket-encryption --bucket example-s3-bucket1
```

Pour plus d'informations et des exemples, consultez get-bucket-encryption dans la Référence des commandes AWS CLI .

Ce qui suit renvoie la configuration des notifications du bucket *example-s3-bucket1*. Pour plus d'informations sur les notifications d'événements du bucket, consultezNotifications d'événements Amazon S3.

```
aws s3api get-bucket-notification-configuration --bucket example-s3-bucket1
```

Pour plus d'informations et des exemples, consultez get-bucket-notification-configuration dans la Référence des commandes AWS CLI .

Ce qui suit renvoie l'état de journalisation du bucket *example-s3-bucket1*. Pour plus d'informations sur la journalisation des compartiments, consultezEnregistrement de demandes avec journalisation des accès au serveur.

```
aws s3api get-bucket-logging --bucket example-s3-bucket1
```

Pour plus d'informations et des exemples, consultez get-bucket-logging dans la Référence des commandes AWS CLI .

Utilisation des AWS SDK

Pour des exemples expliquant comment renvoyer les propriétés d'un bucket avec les AWS SDK, telles que le versionnement, les balises, etc., consultez. Actions pour Amazon S3 à l'aide de AWS kits SDK

Pour des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Vider un compartiment

Vous pouvez vider le contenu d'un compartiment à l'aide de la console Amazon S3, AWS des SDK ou AWS Command Line Interface (AWS CLI). Lorsque vous videz un compartiment, vous supprimez tous les objets, mais vous conservez le compartiment. Après avoir vidé un compartiment, les éléments supprimés ne peuvent être restaurés. Les objets ajoutés au compartiment pendant que l'action est en cours, risquent d'être supprimés. Tous les objets (y compris toutes les versions d'objets et les marqueurs de suppression) du compartiment doivent être supprimés avant que le compartiment lui-même ne puisse être supprimé.

Lorsque vous videz un compartiment dont la gestion des versions S3 est activée ou suspendue, toutes les versions de tous les objets du compartiment sont supprimées. Pour plus d'informations, consultez Utiliser des objets dans un compartiment activé pour la gestion des versions.

Vous pouvez également spécifier une configuration de cycle de vie sur un compartiment pour que les objets expirent afin qu'Amazon S3 puisse les supprimer. Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket. Pour vider un compartiment volumineux, nous vous recommandons d'utiliser une règle de configuration S3 Lifecycle. L'expiration du cycle de vie étant

un processus asynchrone, l'exécution de la règle peut mettre plusieurs jours à vider le compartiment. Après la première exécution de la règle par Amazon S3, tous les objets éligibles à l'expiration sont marqués comme devant être supprimés. Les objets marqués comme devant être supprimés ne vous sont plus facturés. Pour plus d'informations, consultez Comment vider un compartiment Amazon S3 à l'aide d'une règle de configuration du cycle de vie ?.

Utilisation de la console S3

Vous pouvez utiliser la console Amazon S3 pour vider un compartiment, ce qui entraîne la suppression de tous les objets du compartiment, sans supprimer ce dernier.

Pour vider un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, sélectionnez l'option en regard du nom du compartiment à vider, puis choisissez Vider.
3. Sur la page Empty bucket (Vider le compartiment), confirmez que vous souhaitez vider le compartiment en saisissant le nom de ce dernier dans le champ de texte, puis choisissez Empty (Vider).
4. Surveillez la progression du processus de vidage du compartiment sur la page Empty bucket status (Statut du compartiment vide).

À l'aide du AWS CLI

Vous pouvez vider un bucket en utilisant le AWS CLI uniquement si la gestion des versions du bucket n'est pas activée. Si le contrôle de version n'est pas activé, vous pouvez utiliser la AWS CLI commande `rm` (remove) avec le `--recursive` paramètre pour vider le compartiment (ou supprimer un sous-ensemble d'objets avec un préfixe de nom de clé spécifique).

La commande `rm` suivante supprime les objets ayant un préfixe de nom de clé `doc`, par exemple, `doc/doc1` et `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Utilisez la commande suivante pour supprimer tous les objets sans spécifier un préfixe.

```
$ aws s3 rm s3://bucket-name --recursive
```

Pour plus d’informations, consultez Utilisation des commandes S3 de haut niveau avec AWS CLI dans le Guide de l’utilisateur AWS Command Line Interface .

> **Note**
>
> Vous ne pouvez pas supprimer des objets d’un compartiment pour lequel le contrôle de version est activé. Amazon S3 ajoute un marqueur de suppression lorsque vous supprimez un objet, à l’instar de cette commande. Pour plus d’informations sur la gestion des versions des compartiments S3, consultez Utilisation de la gestion des versions dans les compartiments S3.

Utilisation des AWS SDK

Vous pouvez utiliser les AWS SDK pour vider un compartiment ou supprimer un sous-ensemble d'objets dotés d'un préfixe de nom de clé spécifique.

Pour un exemple de la façon de vider un bucket en utilisant AWS SDK for Java, consultezSuppression d'un compartiment. Le code supprime tous les objets, que le compartiment soit activé ou non pour le contrôle de version, puis il supprime le compartiment. Pour simplement vider le compartiment, assurez-vous de supprimer l’instruction de suppression du compartiment.

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez la section Outils pour Amazon Web Services.

Utilisation de la configuration du cycle de vie

Pour vider un compartiment volumineux, nous vous recommandons d'utiliser une règle de configuration S3 Lifecycle. L'expiration du cycle de vie étant un processus asynchrone, l'exécution de la règle peut mettre plusieurs jours à vider le compartiment. Après la première exécution de la règle par Amazon S3, tous les objets éligibles à l'expiration sont marqués comme devant être supprimés. Les objets marqués comme devant être supprimés ne vous sont plus facturés. Pour plus d'informations, consultez Comment vider un compartiment Amazon S3 à l'aide d'une règle de configuration du cycle de vie ?.

Si vous utilisez une configuration de cycle de vie pour vider votre compartiment, la configuration doit inclure des versions actuelles, des versions anciennes, des marqueurs de suppression et des chargements partitionnés incomplets.

Vous pouvez ajouter des règles de configuration du cycle de vie pour que tous les objets ou un sous-ensemble d'objets ayant un préfixe de nom de clé spécifique expirent. Par exemple, pour supprimer tous les objets dans un compartiment, vous pouvez configurer une règle du cycle de vie pour que les objets expirent un jour après leur création.

Amazon S3 prend en charge une règle de cycle de vie de compartiment que vous pouvez utiliser pour arrêter les chargements partitionnés s'ils n'aboutissent pas au bout du nombre de jours spécifié après leur lancement. Nous vous recommandons de configurer cette règle de cycle de vie afin de réduire vos coûts de stockage. Pour plus d'informations, consultez Configuration d'une configuration de cycle de vie de compartiment pour supprimer les chargements partitionnés incomplets.

Pour plus d'informations sur l'utilisation d'une configuration de cycle de vie pour vider un compartiment, consultez Configuration du cycle de vie d'un bucket et Objets en cours d'expiration.

Vidange d'un compartiment avec configuré AWS CloudTrail

AWS CloudTrail suit les événements liés aux données au niveau des objets dans un compartiment Amazon S3, tels que la suppression d'objets. Si vous utilisez un compartiment comme destination pour enregistrer vos CloudTrail événements et que vous supprimez des objets de ce même compartiment, vous créez peut-être de nouveaux objets en vidant votre compartiment. Pour éviter cela, bloquez vos AWS CloudTrail sentiers. Pour plus d'informations sur la façon d'empêcher l'enregistrement d'événements dans vos CloudTrail sentiers, consultez la section Désactiver la journalisation d'un sentier dans le guide de AWS CloudTrail l'utilisateur.

Une autre alternative pour empêcher l'ajout de CloudTrail sentiers au bucket est d'ajouter une `s3:PutObject` déclaration de refus à votre politique de bucket. Si vous souhaitez stocker ultérieurement de nouveaux objets dans le compartiment, vous devrez supprimer cette instruction de refus `s3:PutObject`. Pour plus d'informations, consultez Opérations sur les objets et Éléments de politique JSON IAM : Effect dans le Guide de l'utilisateur IAM.

Suppression d'un compartiment

Vous pouvez supprimer un compartiment Amazon S3 vide. Avant de supprimer un compartiment, tenez compte des points suivants :

- Les noms de compartiment sont uniques. Si vous supprimez un bucket, un autre AWS utilisateur peut utiliser le nom.
- Si le compartiment héberge un site Web statique et que vous avez créé et configuré une zone hébergée Amazon Route 53 comme décrit dans Tutoriel : configuration d'un site Web

statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53, vous devez nettoyer les paramètres de zone hébergée Route 53 qui sont associés au compartiment. Pour plus d'informations, consultez Étape 2 : Supprimer la zone hébergée Route 53.

- Si le compartiment reçoit les données de journaux depuis Elastic Load Balancing (ELB) : nous vous recommandons de cesser la remise des journaux ELB dans ce compartiment avant de le supprimer. Après que vous avez supprimé le compartiment, si un autre utilisateur crée un compartiment à l'aide du même nom, vos données de journaux peuvent potentiellement être remises à ce compartiment. Pour obtenir des informations sur les journaux d'accès ELB, consultez Journaux d'accès dans le Guide de l'utilisateur des équilibreurs de charge classiques et Journaux d'accès dans le Guide de l'utilisateur des équilibreurs de charge des applications.

Résolution des problèmes

Si vous ne parvenez pas à supprimer un compartiment Simple Storage Service (Amazon S3), prenez en compte les points suivants :

- Vérifiez que le compartiment est vide – Vous pouvez uniquement supprimer des compartiments qui ne contiennent aucun objet. Vérifiez que le compartiment est vide.
- Assurez-vous qu'il n'y a pas de points d'accès attachés : vous ne pouvez supprimer que les compartiments auxquels aucun point d'accès n'est attaché. Supprimez tous les points d'accès qui sont attachés au compartiment, avant de supprimer le compartiment.
- AWS Organizations politiques de contrôle des services (SCP) : une politique de contrôle des services peut refuser l'autorisation de suppression d'un bucket. Pour obtenir des informations sur les SCP, consultez Politiques de contrôle des services dans le Guide de l'utilisateur AWS Organizations .
- s3 : DeleteBucket autorisations — Si vous ne pouvez pas supprimer un compartiment, contactez votre administrateur IAM pour confirmer que vous disposez des `s3:DeleteBucket` autorisations. Pour plus d'informations sur l'affichage ou la mise à jour des autorisations IAM, veuillez consulter Modification des autorisations pour un utilisateur IAM dans le Guide de l'utilisateur IAM.
- s3 : déclaration de DeleteBucket refus — Si vous disposez d'`s3:DeleteBucket`autorisations dans votre politique IAM et que vous ne pouvez pas supprimer un compartiment, la politique de compartiment peut inclure une instruction de refus pour`s3:DeleteBucket`. Les compartiments créés par ElasticBeanstalk ont une politique contenant cette instruction par défaut. Avant de pouvoir supprimer le compartiment, vous devez supprimer cette instruction ou la stratégie de compartiment.

> **Important**
>
> Les noms de compartiment sont uniques. Si vous supprimez un bucket, un autre AWS utilisateur peut utiliser le nom. Si vous souhaitez réutiliser le même nom de compartiment, ne supprimez pas le compartiment. Nous vous recommandons de vider le compartiment et de le conserver.

Utiliser la console S3.

Pour supprimer un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste des Buckets (Compartiments), sélectionnez l'option en regard du nom du compartiment à supprimer, puis choisissez Delete (Supprimer) en haut de la page.
3. Dans la page Delete bucket (Supprimer le compartiment), confirmez que vous souhaitez supprimer le compartiment en saisissant le nom de ce dernier dans le champ de texte, puis choisissez Delete bucket (Supprimer le compartiment).

   > **Note**
   >
   > Si le compartiment contient des objets, videz-le avant de le supprimer en sélectionnant le lien de configuration du compartiment vide dans l'alerte d'erreur This bucket is not empty (Ce compartiment n'est pas vide) et en suivant les instructions de la page Empty bucket (Compartiment vide). Revenez ensuite à la page Delete bucket (Supprimer le compartiment) et supprimez le compartiment.

4. Pour vérifier que vous avez supprimé le compartiment, ouvrez la liste Buckets (Compartiments) et saisissez le nom du compartiment que vous avez supprimé. Si le compartiment est introuvable, votre suppression a réussi.

Utilisation du AWS SDK pour Java

L'exemple suivant montre comment supprimer un bucket à l'aide du AWS SDK for Java. Tout d'abord, le code supprime les objets du compartiment puis il supprime le compartiment. Pour en savoir plus sur les autres kits SDK AWS , consultez Outils pour Amazon Web Services.

Java

L'exemple Java suivant supprime un compartiment qui contient des objets. Il supprime tous les objets, puis il supprime le compartiment. L'exemple fonctionne pour les compartiments avec ou sans la gestion des versions activée.

> **Note**
>
> Pour les compartiments sans la gestion des versions activée, vous pouvez supprimer tous les objets directement, puis supprimer le compartiment. Pour les compartiments avec la gestion des versions activée, vous devez supprimer toutes les versions d'objet avant de supprimer le compartiment.

Pour obtenir des instructions sur la création et le test d'un échantillon de travail, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.Iterator;

public class DeleteBucket2 {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
```

```
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Delete all objects from the bucket. This is sufficient
            // for unversioned buckets. For versioned buckets, when you attempt to
delete
            // objects, Amazon S3 inserts
            // delete markers for all objects, but doesn't delete the object
versions.
            // To delete objects from versioned buckets, delete all of the object
versions
            // before deleting
            // the bucket (see below for an example).
            ObjectListing objectListing = s3Client.listObjects(bucketName);
            while (true) {
                Iterator<S3ObjectSummary> objIter =
objectListing.getObjectSummaries().iterator();
                while (objIter.hasNext()) {
                    s3Client.deleteObject(bucketName, objIter.next().getKey());
                }

                // If the bucket contains many objects, the listObjects() call
                // might not return all of the objects in the first listing. Check
to
                // see whether the listing was truncated. If so, retrieve the next
page of
                // objects
                // and delete them.
                if (objectListing.isTruncated()) {
                    objectListing = s3Client.listNextBatchOfObjects(objectListing);
                } else {
                    break;
                }
            }

            // Delete all object versions (required for versioned buckets).
            VersionListing versionList = s3Client.listVersions(new
ListVersionsRequest().withBucketName(bucketName));
            while (true) {
                Iterator<S3VersionSummary> versionIter =
versionList.getVersionSummaries().iterator();
                while (versionIter.hasNext()) {
```

```
                    S3VersionSummary vs = versionIter.next();
                    s3Client.deleteVersion(bucketName, vs.getKey(),
 vs.getVersionId());
                }

                if (versionList.isTruncated()) {
                    versionList = s3Client.listNextBatchOfVersions(versionList);
                } else {
                    break;
                }
            }

            // After all objects and object versions are deleted, delete the bucket.
            s3Client.deleteBucket(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
 couldn't
            // parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

À l'aide du AWS CLI

Vous pouvez supprimer un compartiment contenant des objets avec le AWS CLI si le contrôle de version n'est pas activé. Quand vous supprimez un compartiment qui contient des objets, tous les objets du compartiment sont définitivement supprimés, y compris ceux qui sont transmis à la classe de stockage `S3 Glacier`.

Si la gestion des versions n'est pas activée dans votre bucket, vous pouvez utiliser la AWS CLI commande `rb` (remove bucket) avec le `--force` paramètre pour supprimer le bucket et tous les objets qu'il contient. Cette commande supprime d'abord tous les objets, puis elle supprime le compartiment.

Si la gestion des versions est activée, les objets versionnés ne sont pas supprimés dans ce processus, ce qui entraînerait l'échec de la suppression du compartiment car ce dernier ne serait pas vide. Pour obtenir des informations sur la suppression d'objets versionnés, consultez Suppression des versions d'objet.

```
$ aws s3 rb s3://bucket-name --force
```

Pour plus d'informations, consultez la section Utilisation de commandes S3 de haut niveau AWS Command Line Interface dans le guide de AWS Command Line Interface l'utilisateur.

Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Tous les compartiments Amazon S3 ont le chiffrement configuré par défaut et les objets sont automatiquement chiffrés à l'aide du chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3). Ce paramètre de chiffrement s'applique à tous les objets de vos compartiments Amazon S3.

Si vous avez besoin d'un contrôle accru sur vos clés, par exemple pour gérer la rotation des clés et les autorisations de politique d'accès, vous pouvez choisir d'utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou le chiffrement côté serveur à double couche avec clés (DSSE-KMS). AWS KMS Pour en savoir plus sur la modification des clés KMS, consultez Modification des clés dans le Guide du développeur AWS Key Management Service .

> **Note**
>
> Nous avons modifié les compartiments afin de chiffrer automatiquement les nouveaux chargements d'objets. Si vous avez déjà créé un compartiment sans chiffrement par défaut, Amazon S3 activera le chiffrement par défaut pour le compartiment à l'aide de SSE-S3. Aucune modification ne sera apportée à la configuration de chiffrement par défaut d'un compartiment existant pour lequel le chiffrement SSE-S3 ou SSE-KMS est déjà configuré. Si vous souhaitez chiffrer vos objets avec SSE-KMS, vous devez modifier le type de chiffrement dans les paramètres de votre compartiment. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

Lorsque vous configurez votre compartiment pour utiliser le chiffrement par défaut avec SSE-KMS, vous pouvez également activer les clés de compartiment S3 afin de réduire le trafic de demandes en provenance d'Amazon S3 AWS KMS et de réduire le coût du chiffrement. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Pour identifier les compartiments dont le chiffrement par défaut est activé par SSE-KMS, vous pouvez utiliser les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Using S3 Storage Lens to protect your data (Utilisation de S3 Storage Lens pour protéger vos données).

Lorsque vous utilisez le chiffrement côté serveur, Amazon S3 chiffre un objet avant de l'enregistrer sur disque et le déchiffre lorsque vous téléchargez l'objet. Pour plus d'informations sur la protection des données à l'aide du chiffrement côté serveur et de la gestion des clés de chiffrement, consultez Protection des données avec le chiffrement côté serveur.

Pour en savoir plus sur les autorisations nécessaires pour le chiffrement par défaut, consultez PutBucketEncryption dans la documentation de référence de l'API Amazon Simple Storage Service.

Vous pouvez configurer le comportement de chiffrement par défaut d'Amazon S3 pour un compartiment S3 à l'aide de la console Amazon S3, AWS des SDK, de l'API REST Amazon S3 et de l'interface de ligne de AWS commande (AWS CLI).

Chiffrement des objets existants

Pour chiffrer vos objets Amazon S3 non chiffrés existants, vous pouvez utiliser des opérations par lot Amazon S3. Vous fournissez à la fonctionnalité d'opérations par lots S3 une liste d'objets sur lesquels agir. La fonctionnalité d'opérations par lots appelle l'API correspondante pour exécuter l'opération spécifiée. Vous pouvez utiliser l'opération de copie des opérations par lot pour copier des objets non chiffrés et les réécrire dans le même compartiment en tant qu'objets chiffrés. Une tâche d'opérations par lots peut effectuer l'opération spécifiée sur des milliards d'objets. Pour plus d'informations, consultez Exécution des opérations par lot à grande échelle sur des objets Amazon S3 et le billet de blog sur le stockage AWS intitulé Encrypting objects with Amazon S3 Batch Operations.

Vous pouvez également chiffrer des objets existants à l'aide de l'opération `CopyObject` API ou de la `copy-object` AWS CLI commande. Pour plus d'informations, consultez le billet de blog sur le stockage AWS intitulé Encrypting existing Amazon S3 objects with the AWS CLI.

> **Note**
>
> Les compartiments Amazon S3 pour lesquels le chiffrement de compartiment par défaut est défini sur SSE-KMS ne peuvent pas servir de compartiments de destination pour the section called "Enregistrement de l'accès au serveur". Seul le chiffrement par défaut SSE-S3 est pris en charge pour les compartiments de destination du journal d'accès au serveur.

Utilisation du chiffrement SSE-KMS pour les opérations intercomptes

Tenez compte des éléments suivants lors de l'utilisation du chiffrement pour les opérations inter-comptes :

- Si aucun nom de ressource AWS KMS key Amazon (ARN) ou alias n'est fourni au moment de la demande ou via la configuration de chiffrement par défaut du bucket, le Clé gérée par AWS (`aws/s3`) est utilisé.
- Si vous téléchargez ou accédez à des objets S3 à l'aide de principes AWS Identity and Access Management (IAM) identiques Compte AWS à ceux de votre clé KMS, vous pouvez utiliser le Clé gérée par AWS (). `aws/s3`
- Utilisez une clé gérée par le client si vous souhaitez accorder un accès intercompte à vos objets S3. Vous pouvez configurer la politique d'une clé gérée par le client afin d'autoriser l'accès à partir d'un autre compte.
- Si vous spécifiez une clé KMS gérée par le client, nous vous recommandons d'utiliser un ARN de clé KMS entièrement qualifié. Si vous utilisez plutôt un alias de clé KMS, AWS KMS la clé est

résolue dans le compte du demandeur. En raison de ce comportement, les données peuvent être chiffrées avec une clé KMS qui appartient au demandeur, et non au propriétaire du compartiment.

- Vous devez spécifier une clé pour laquelle vous (le demandeur) avez obtenu l'autorisation de `Encrypt`. Pour en savoir plus, consultez Permettre aux utilisateurs de clés d'utiliser une clé KMS pour les opérations de chiffrement dans le Guide de l'utilisateur AWS Key Management Service .

Pour plus d'informations sur les circonstances dans lesquelles utiliser des clés gérées par le client et des clés KMS AWS gérées, consultez Dois-je utiliser une clé gérée par le client Clé gérée par AWS ou une clé gérée par le client pour chiffrer mes objets dans Amazon S3 ?

Utilisation du chiffrement par défaut avec la réplication

Après avoir activé le chiffrement par défaut pour un compartiment de destination de réplication, le comportement de chiffrement suivant s'applique :

- Si des objets du compartiment source ne sont pas chiffrés, les objets réplica du compartiment de destination sont chiffrés à l'aide des paramètres de chiffrement par défaut du compartiment de destination. Par conséquent, les balises d'entité (ETags) des objets sources diffèrent des ETags des objets réplica. Si certaines de vos applications utilisent des ETags, vous devez les mettre à jour pour tenir compte de cette différence.
- Si les objets du compartiment source sont chiffrés à l'aide d'un chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3), d'un chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou d'un chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS), les objets répliques du compartiment de destination utilisent le même type de chiffrement que les objets source. Les paramètres de chiffrement par défaut du compartiment de destination ne sont pas utilisés.

Pour plus d’informations sur l’utilisation du chiffrement par défaut avec SSE-KMS, consultez Réplication d'objets chiffrés.

Utilisation des clés de compartiment Amazon S3 avec chiffrement par défaut

Lorsque vous configurez votre compartiment pour utiliser SSE-KMS comme le comportement de chiffrement par défaut sur de nouveaux objets, vous pouvez également configurer des clés de compartiment S3. Les clés de compartiment S3 réduisent le nombre de transactions depuis Amazon S3 AWS KMS afin de réduire le coût du SSE-KMS.

Lorsque vous configurez votre compartiment pour utiliser les clés de compartiment S3 pour SSE-KMS sur de nouveaux objets, il AWS KMS génère une clé au niveau du compartiment qui est utilisée pour créer une clé de données unique pour les objets du compartiment. Cette clé de compartiment S3 est utilisée pendant une période limitée dans le temps dans Amazon S3, ce qui réduit la nécessité pour Amazon S3 de faire des demandes AWS KMS pour effectuer des opérations de chiffrement.

Pour plus d'informations sur l'utilisation des clés de compartiment S3, consultez Utilisation de clés de compartiment Amazon S3..

Configuration du chiffrement par défaut

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Les compartiments Amazon S3 ont le chiffrement des compartiments activé par défaut et les nouveaux objets sont automatiquement chiffrés à l'aide du chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3). Ce chiffrement s'applique à tous les nouveaux objets de vos compartiments Amazon S3, sans frais.

Si vous avez besoin d'un contrôle accru sur vos clés de chiffrement, par exemple pour gérer la rotation des clés et l'attribution des politiques d'accès, vous pouvez choisir d'utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou le chiffrement côté serveur à double couche avec clés (DSSE-KMS). AWS KMS Pour en savoir plus sur SSE-KMS, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS). Pour en savoir plus sur DSSE-KMS, consultez the section called “Chiffrement double couche côté serveur (DSSE-KMS)”.

Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez avoir l'autorisation d'utiliser la clé. Pour plus d'informations sur les autorisations intercomptes pour les clés

KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service .

Lorsque vous définissez le chiffrement de compartiment par défaut sur SSE-KMS, vous pouvez également configurer une clé de compartiment S3 afin de réduire les coûts de vos AWS KMS demandes. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

> **Note**
>
> Si vous avez l'habitude PutBucketEncryptionde définir le chiffrement de votre compartiment par défaut sur SSE-KMS, vous devez vérifier que l'ID de votre clé KMS est correct. Amazon S3 ne valide pas l'ID de clé KMS fourni dans les PutBucketEncryption demandes.

Il n'y a pas de frais supplémentaires relatifs à l'utilisation du chiffrement par défaut pour les compartiments S3. Les demandes de configuration du comportement de chiffrement par défaut seront facturées comme des demandes Amazon S3 standard. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3. Pour le SSE-KMS et le DSSE-KMS, des AWS KMS frais s'appliquent et sont indiqués dans le prix.AWS KMS

Le chiffrement côté serveur avec des clés fournies par le client (SSE-C) n'est pas pris en charge pour le chiffrement par défaut.

Vous pouvez configurer le chiffrement par défaut d'Amazon S3 pour un compartiment S3 à l'aide de la console Amazon S3, AWS des SDK, de l'API REST Amazon S3 et du AWS Command Line Interface (AWS CLI).

Modifications à prendre en compte avant d'activer le chiffrement par défaut

Après avoir activé le chiffrement par défaut pour un compartiment, le comportement de chiffrement suivant s'applique :

- Il n'y a pas de modification pour le chiffrement des objets qui existaient dans le compartiment avant l'activation du chiffrement par défaut.
- Lorsque vous chargez des objets après l'activation du chiffrement par défaut :
  - Si vos en-têtes de demandes PUT ne comportent pas d'informations de chiffrement, Amazon S3 utilise les paramètres de chiffrement par défaut du compartiment pour chiffrer les objets.

- Si vos en-têtes de demandes PUT comportent des informations de chiffrement, Amazon S3 utilise les informations de la demande PUT pour chiffrer les objets avant de les stocker dans Amazon S3.

- Si vous utilisez l'option SSE-KMS ou DSSE-KMS pour votre configuration du chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les quotas de AWS KMS et sur la procédure à suivre pour demander une augmentation des quotas, consultez Quotas dans le Guide du développeur AWS Key Management Service .

> **Note**
>
> Les objets chargés avant l'activation du chiffrement par défaut ne seront pas chiffrés. Pour plus d'informations sur le chiffrement d'objets existants, consultez the section called “Définition du chiffrement du compartiment par défaut”.

Utilisation de la console S3

Pour configurer le chiffrement par défaut sur un compartiment Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment que vous souhaitez utiliser.
4. Choisissez l’onglet Propriétés.
5. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).
6. Pour configurer le chiffrement, sous Type de chiffrement, choisissez l'une des options suivantes :

   - Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
   - Chiffrement côté serveur avec AWS Key Management Service clés (SSE-KMS)
   - Chiffrement double couche côté serveur avec AWS Key Management Service clés (DSSE-KMS)

> **Important**
>
> Si vous utilisez l'option SSE-KMS ou DSSE-KMS pour votre configuration du chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les AWS KMS quotas et sur la manière de demander une augmentation de quota, consultez la section Quotas dans le guide du AWS Key Management Service développeur.

Les compartiments et les nouveaux objets sont chiffrés par défaut avec SSE-S3, sauf si vous spécifiez un autre type de chiffrement par défaut pour vos compartiments. Pour plus d'informations sur le chiffrement par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Pour en savoir plus sur l'utilisation du chiffrement côté serveur Amazon S3 pour chiffrer vos données, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

7. Si vous avez choisi le chiffrement côté serveur avec AWS Key Management Service clés (SSE-KMS) ou le chiffrement côté serveur double couche avec AWS Key Management Service clés (DSSE-KMS), procédez comme suit :

   a. Sous CléAWS KMS , spécifiez votre clé KMS de l'une des manières suivantes :

      - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

        La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez la section Clés et AWS clés client dans le Guide du AWS Key Management Service développeur.

      - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.

      - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

        Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous ne pouvez utiliser que les clés KMS activées au même endroit Région AWS que le bucket. Lorsque vous choisissez Choisissez une clé parmi vos clés KMS, la console S3 ne répertorie que 100 clés KMS par Région. Si vous avez plus de 100 clés KMS dans la même Région, vous ne pourrez voir que les 100 premières clés KMS dans la console S3. Pour utiliser une clé KMS qui ne figure pas dans la console, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de la clé KMS.
>
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon WorkMail prend uniquement en charge que les clés KMS de chiffrement symétriques. Pour plus d'informations sur ces clés, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur l'utilisation de SSE-KMS avec Amazon S3, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS). Pour en savoir plus sur l'utilisation de DSSE-KMS, consultez the section called “Chiffrement double couche côté serveur (DSSE-KMS)”.

b. Lorsque vous configurez votre compartiment pour utiliser le chiffrement par défaut avec SSE-KMS, vous pouvez également activer la clé de compartiment S3. Les clés de compartiment S3 réduisent le coût du chiffrement en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d’informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

   Pour utiliser les clés de compartiment S3, sous la Clé de compartiment, choisissez Activer.

   > **Note**
   >
   > Les clés de compartiment S3 ne sont pas prises en charge pour DSSE-KMS.

8. Sélectionnez Enregistrer les modifications.

À l'aide du AWS CLI

Ces exemples montrent comment configurer le chiffrement par défaut en utilisant SSE-S3 ou SSE-KMS avec une clé de compartiment S3.

Pour plus d'informations sur le chiffrement par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3. Pour plus d'informations sur l'utilisation du AWS CLI pour configurer le chiffrement par défaut, consultez put-bucket-encryption.

Example — Chiffrement par défaut avec SSE-S3

Cet exemple montre comment configurer le chiffrement du compartiment par défaut avec les clés gérées par Amazon S3.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example — Chiffrement par défaut avec SSE-KMS à l'aide d'une clé de compartiment S3

Cet exemple configure le chiffrement du compartiment par défaut avec SSE-KMS à l'aide d'une clé de compartiment S3.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
```

```
    }'
```

Utilisation de l'API REST

Utilisez l'opération `PutBucketEncryption` de l'API REST pour activer le chiffrement par défaut et définir le type de chiffrement côté serveur à utiliser : SSE-S3, SSE-KMS ou DSSE-KMS.

Pour plus d'informations, veuillez consulter PutBucketEncryption dans la Référence d'API Amazon Simple Storage Service.

Surveillance du chiffrement par défaut avec Amazon AWS CloudTrail et Amazon EventBridge

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Vous pouvez suivre les demandes de configuration de chiffrement par défaut pour les compartiments Amazon S3 à l'aide d'événements AWS CloudTrail . Les noms d'événements d'API suivants sont utilisés dans CloudTrail les journaux :

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

Vous pouvez également créer des EventBridge règles correspondant aux CloudTrail événements de ces appels d'API. Pour plus d'informations sur CloudTrail les événements, consultezActiver la journalisation des objets d'un compartiment à l'aide de la console. Pour plus d'informations sur les EventBridge événements, consultez la section Événements de Services AWS.

Vous pouvez utiliser CloudTrail les journaux pour les actions Amazon S3 au niveau de l'objet à suivre PUT et pour les POST demandes adressées à Amazon S3. Vous pouvez utiliser ces actions pour vérifier si le chiffrement par défaut est utilisé pour chiffrer des objets lorsque les demandes PUT entrantes n'ont pas d'en-têtes de chiffrement.

Quand Amazon S3 chiffre un objet selon les paramètres de chiffrement par défaut, le journal inclut l'un des champs suivants comme paire nom-valeur : `"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"` ou `"SSEApplied":"Defa`

Quand Amazon S3 chiffre un objet en utilisant les en-têtes de chiffrement PUT, le journal inclut l'un des champs suivants comme paire nom-valeur : `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"` ou `"SSEApplied":"SSE_C"`.

Pour les chargements en plusieurs parties, cette information est incluse dans vos demandes d'opération d'API `InitiateMultipartUpload`. Pour plus d'informations sur l'utilisation CloudTrail de et CloudWatch, consultezSurveillance d'Amazon S3.

Utilisation de Mountpoint pour Amazon S3

Mountpoint pour Amazon S3 est un client de fichiers open source à haut débit permettant de monter un compartiment Amazon S3 en tant que système de fichiers local. Avec Mountpoint, vos applications peuvent accéder aux objets stockés dans Amazon S3 par le biais d'opérations de système de fichiers, telles qu'ouvrir et lire. Mountpoint traduit automatiquement ces opérations en appels d'API d'objet S3, permettant à vos applications d'accéder au stockage et au débit élastiques d'Amazon S3 via une interface de fichier.

Mountpoint pour Amazon S3 est généralement disponible pour une utilisation en production sur vos applications à lecture intensive à grande échelle : lacs de données, entraînement de machine learning, rendu d'images, simulation de véhicules autonomes, extraction, transformation et chargement (ETL), etc.

Mountpoint prend en charge les opérations de base de système de fichiers et peut lire des fichiers d'une taille maximale de 5 To. Il peut répertorier et lire des fichiers existants, et il peut en créer de nouveaux. Il ne peut pas modifier les fichiers existants ni supprimer de répertoires, et il ne prend pas en charge les liens symboliques ni le verrouillage de fichiers. Mountpoint est idéal pour les applications qui n'ont pas besoin de toutes les fonctionnalités d'un système de fichiers partagé et des autorisations de style POSIX, mais qui ont besoin du débit élastique d'Amazon S3 pour lire et écrire de grands jeux de données S3. Pour obtenir des détails, consultez Comportement du système

de fichiers Mountpoint (langue française non garantie) sur GitHub. Pour les charges de travail qui nécessitent une prise en charge POSIX complète, nous recommandons Amazon FSx pour Lustre et sa prise en charge de la liaison des compartiments S3.

Mountpoint pour Amazon S3 est disponible uniquement pour les systèmes d'exploitation Linux. Vous pouvez utiliser Mountpoint pour accéder aux objets S3 dans toutes les classes de stockage, sauf S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, le niveau Archive Access de S3 Intelligent-Tiering et le niveau Deep Archive Access de S3 Intelligent-Tiering.

Rubriques

Installation de Mountpoint

Vous pouvez télécharger et installer des packages précréés de Mountpoint pour Amazon S3 à l'aide de la ligne de commande. Les instructions de téléchargement et d'installation de Mountpoint varient en fonction du système d'exploitation Linux que vous utilisez.

Rubriques

Distributions basées sur RPM (Amazon Linux, Fedora, CentOS, RHEL)

1. Copiez l'URL de téléchargement suivante pour votre architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm
   ```

2. Téléchargez le package Mountpoint pour Amazon S3. Remplacez *download-link* par l'URL de téléchargement appropriée indiquée à l'étape précédente.

```
wget download-link
```

3. (Facultatif) Vérifiez l'authenticité et l'intégrité du fichier téléchargé. Tout d'abord, copiez l'URL de signature appropriée pour votre architecture.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm.asc
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm.asc
```

   Ensuite, consultez Vérification de la signature du package Mountpoint pour Amazon S3.

4. Installez le package à l'aide de la commande suivante :

```
sudo yum install ./mount-s3.rpm
```

5. Vérifiez que Mountpoint est correctement installé en saisissant la commande suivante :

```
mount-s3 --version
```

   Vous devez voir des résultats similaires à ce qui suit :

```
mount-s3 1.3.1
```

Distributions basées sur DEB (Debian, Ubuntu)

1. Copiez l'URL de téléchargement pour votre architecture.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb
```

2. Téléchargez le package Mountpoint pour Amazon S3. Remplacez *download-link* par l'URL de téléchargement appropriée indiquée à l'étape précédente.

```
wget download-link
```

3. (Facultatif) Vérifiez l'authenticité et l'intégrité du fichier téléchargé. Tout d'abord, copiez l'URL de signature pour votre architecture.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb.asc
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb.asc
```

   Ensuite, consultez Vérification de la signature du package Mountpoint pour Amazon S3.

4. Installez le package à l'aide de la commande suivante :

```
sudo apt-get install ./mount-s3.deb
```

5. Vérifiez que Mountpoint pour Amazon S3 est correctement installé en exécutant la commande suivante :

```
mount-s3 --version
```

   Vous devez voir des résultats similaires à ce qui suit :

```
mount-s3 1.3.1
```

Autres distributions Linux

1. Consultez la documentation de votre système d'exploitation pour installer les packages FUSE et `libfuse2`, qui sont obligatoires.

2. Copiez l'URL de téléchargement pour votre architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz
   ```

3. Téléchargez le package Mountpoint pour Amazon S3. Remplacez *download-link* par l'URL de téléchargement appropriée indiquée à l'étape précédente.

   ```
   wget download-link
   ```

4. (Facultatif) Vérifiez l'authenticité et l'intégrité du fichier téléchargé. Tout d'abord, copiez l'URL de signature pour votre architecture.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz.asc
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz.asc
   ```

   Ensuite, consultez Vérification de la signature du package Mountpoint pour Amazon S3.

5. Installez le package à l'aide de la commande suivante :

   ```
   sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
   mount-s3.tar.gz
   ```

6. Ajoutez le fichier binaire `mount-s3` à votre variable d'environnement PATH. Dans votre fichier `$HOME/.profile`, ajoutez la ligne suivante :

   ```
   export PATH=$PATH:/opt/aws/mountpoint-s3/bin
   ```

   Enregistrez le fichier `.profile` et exécutez la commande suivante :

```
source $HOME/.profile
```

7. Vérifiez que Mountpoint pour Amazon S3 est correctement installé en exécutant la commande suivante :

   ```
   mount-s3 --version
   ```

   Vous devez voir des résultats similaires à ce qui suit :

   ```
   mount-s3 1.3.1
   ```

Vérification de la signature du package Mountpoint pour Amazon S3

1. Installez GnuPG (commande gpg). Il est nécessaire de vérifier l'authenticité et l'intégrité d'un package Mountpoint pour Amazon S3 téléchargé. GnuPG est installé par défaut sur des Amazon Machine Images (AMI) Amazon Linux. Après avoir installé GnuPG, passez à l'étape 2.
2. Téléchargez la clé publique Mountpoint en exécutant la commande suivante :

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```

3. Importez la clé publique Mountpoint dans votre trousseau de clés en exécutant la commande suivante :

   ```
   gpg --import KEYS
   ```

4. Vérifiez l'empreinte digitale de la clé publique Mountpoint en exécutant la commande suivante :

   ```
   gpg --fingerprint mountpoint-s3@amazon.com
   ```

   Confirmez que la chaîne d'empreinte digitale affichée correspond à ce qui suit :

   ```
   673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
   ```

   Si la chaîne d'empreinte digitale ne correspond pas, ne terminez pas l'installation de Mountpoint et contactez AWS Support.

5. Téléchargez le fichier de signature de package. Remplacez *signature-link* par le lien de signature appropriée provenant des sections précédentes.

```
wget signature-link
```

6. Vérifiez la signature du package téléchargé en exécutant la commande suivante. Remplacez *signature-filename* par le nom de fichier provenant de l'étape précédente.

```
gpg --verify signature-filename
```

   Par exemple, dans les distributions basées sur RPM, y compris Amazon Linux, saisissez la commande suivante :

```
gpg --verify mount-s3.rpm.asc
```

7. La sortie doit inclure l'expression `Good signature`. Si la sortie inclut l'expression `BAD signature`, téléchargez à nouveau le fichier du package Mountpoint et répétez ces étapes. Si le problème persiste, ne terminez pas l'installation de Mountpoint et contactez AWS Support.

   La sortie peut inclure un avertissement concernant une signature approuvée. Cela n'indique pas qu'il y a un problème. Cela signifie simplement que vous n'avez pas vérifié de manière indépendante la clé publique Mountpoint.

Configuration et utilisation de Mountpoint

Pour utiliser Mountpoint pour Amazon S3, votre hôte a besoin AWS d'informations d'identification valides lui permettant d'accéder au ou aux compartiments que vous souhaitez monter. Pour connaître les différentes méthodes d'authentification, consultez les informations d'identification AWS de Mountpoint sur GitHub.

Par exemple, vous pouvez créer un nouvel utilisateur AWS Identity and Access Management (IAM) et un nouveau rôle à cette fin. Assurez-vous que ce rôle a accès au ou aux compartiments que vous souhaitez monter. Vous pouvez transmettre le rôle IAM à votre instance Amazon EC2 avec un profil d'instance.

Utilisation de Mountpoint pour Amazon S3

Utilisez Mountpoint pour Amazon S3 pour effectuer les opérations suivantes :

1. Montez les compartiments avec la commande `mount-s3`.

   Dans l'exemple suivant, remplacez *DOC-EXAMPLE-BUCKET* par le nom de votre compartiment S3 et remplacez *~/mnt* par le répertoire situé sur votre hôte, où vous souhaitez que votre compartiment S3 soit monté.

   ```
   mkdir ~/mnt
   mount-s3 DOC-EXAMPLE-BUCKET ~/mnt
   ```

   Comme le client Mountpoint s'exécute en arrière-plan par défaut, le répertoire *~/mnt* vous donne désormais accès aux objets de votre compartiment S3.

2. Accédez aux objets dans votre compartiment via Mountpoint.

   Après avoir monté votre compartiment localement, vous pouvez utiliser les commandes Linux courantes, telles que `cat` et `ls`, pour manipuler vos objets S3. Mountpoint pour Amazon S3 interprète les clés de votre compartiment S3 comme des chemins de système de fichiers en les divisant au niveau du caractère de barre oblique (`/`). Par exemple, si vous avez la clé d'objet `Data/2023-01-01.csv` dans votre compartiment, vous aurez un répertoire nommé `Data`dans votre système de fichiers Mountpoint, avec un fichier nommé `2023-01-01.csv` à l'intérieur.

   Mountpoint pour Amazon S3, volontairement, n'implémente pas l'intégralité de la spécification standard POSIX pour les systèmes de fichiers. Mountpoint est optimisé pour les charges de travail qui nécessitent un accès haut débit en lecture et en écriture aux données stockées dans Amazon S3 via une interface de système de fichiers, mais qui ne dépendent pas des fonctionnalités du système de fichiers. Pour plus d'informations, consultez le comportement du système de fichiers de Mountpoint pour Amazon S3 sur GitHub. Les clients qui ont besoin d'une sémantique de système de fichiers plus riche devraient envisager d'autres services de AWS fichiers, tels qu'Amazon Elastic File System (Amazon EFS) ou Amazon FSx.

3. Démontez votre compartiment à l'aide de la commande `umount`. Cette commande démonte votre compartiment S3 et quitte Mountpoint.

   Pour utiliser l'exemple de commande suivant, remplacez *~/mnt* par le répertoire situé sur votre hôte, où votre compartiment S3 est monté.

   ```
   umount ~/mnt
   ```

> **Note**
>
> Pour obtenir la liste des options pour cette commande, exécutez `umount --help`.

Pour plus de détails sur la configuration de Mountpoint, consultez la configuration des compartiment S3 et la configuration du système de fichiers sur GitHub.

Configuration de la mise en cache dans Mountpoint

Lorsque vous utilisez Mountpoint pour Amazon S3, vous pouvez le configurer pour mettre en cache les données les plus récemment consultées depuis vos compartiments S3 sur le stockage d'instance Amazon EC2 ou sur un volume Amazon EBS attaché. La mise en cache de ces données peut contribuer à améliorer les performances et à réduire le coût des accès répétés aux données. La mise en cache dans Mountpoint est idéale pour les cas d'utilisation où vous lisez à plusieurs reprises les mêmes données qui ne changent pas au cours des multiples lectures. Par exemple, vous pouvez utiliser la mise en cache avec des tâches d'entraînement de machine learning qui nécessitent de lire plusieurs fois un jeu de données d'entraînement pour améliorer la précision du modèle.

Lorsque vous montez un compartiment S3, vous pouvez éventuellement activer la mise en cache via des indicateurs. Vous pouvez configurer l'emplacement et la taille du cache de données, ainsi que la durée pendant laquelle les métadonnées sont retenues dans le cache. Lorsque vous montez un compartiment et que la mise en cache est activée, Mountpoint crée un sous-répertoire vide à l'emplacement de cache configuré, si ce sous-répertoire n'existe pas déjà. Lorsque vous montez un compartiment pour la première fois et que vous le démontez, Mountpoint supprime le contenu de l'emplacement du cache. Pour plus d'informations sur la configuration et l'utilisation de la mise en cache dans Mountpoint, consultez la section Configuration de la mise en cache de Mountpoint pour Amazon S3 sur. GitHub

Lorsque vous montez un compartiment S3, vous pouvez activer la mise en cache à l'aide de l'indicateur `--cache` *`CACHE_PATH`*. Dans l'exemple suivant, remplacez *`CACHE_PATH`* par le chemin d'accès au répertoire dans lequel vous souhaitez mettre en cache les données. Remplacez *`DOC-EXAMPLE-BUCKET`* par le nom de votre compartiment S3 et remplacez *`~/mnt`* par le répertoire situé sur votre hôte, où vous souhaitez que votre compartiment S3 soit monté.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH DOC-EXAMPLE-BUCKET ~/mnt
```

> **Important**
>
> Si vous activez la mise en cache, Mountpoint conservera le contenu des objets non chiffrés de votre compartiment S3 à l'emplacement de mise en cache configuré lors du montage. Afin de protéger vos données, nous vous recommandons de restreindre l'accès à l'emplacement du cache de données.

Résolution des problèmes liés à Mountpoint

Mountpoint pour Amazon S3 est soutenu par. AWS Support Si vous avez besoin d'aide, contactez le Centre de AWS Support.

Vous pouvez également passer en revue et soumettre des problèmes Mountpoint sur GitHub.

Si vous découvrez un problème de sécurité potentiel dans ce projet, nous vous demandons de le signaler à AWS Security via notre page de signalement des vulnérabilités. Ne créez pas de problème GitHub public.

Si votre application se comporte de manière inattendue avec Mountpoint, vous pouvez consulter les informations de votre journal pour diagnostiquer le problème.

Journalisation

Par défaut, Mountpoint émet des informations de journal de haute gravité dans syslog.

Pour consulter les journaux dans la plupart des distributions Linux modernes, y compris Amazon Linux, exécutez la commande `journald` suivante :

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Sur les autres systèmes Linux, les entrées `syslog` sont probablement écrites dans un fichier tel que `/var/log/syslog`.

Vous pouvez utiliser ces journaux pour dépanner votre application. Par exemple, si votre application tente de remplacer un fichier existant, l'opération échoue et vous verrez une ligne similaire à la suivante dans le journal :

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full
 key "README.md") is not writable
```

Pour plus d'informations, consultez Journalisation (langue française non garantie) de Mountpoint pour Amazon S3 sur GitHub.

Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration

La fonctionnalité au niveau du compartiment Amazon S3 Transfer Acceleration permet un transfert rapide, facile et sécurisé de fichiers sur des longues distances entre votre client et un compartiment S3. Transfer Acceleration est conçu pour optimiser les vitesses de transfert dans le monde entier dans des compartiments S3. Transfer Acceleration tire parti des emplacements périphériques répartis dans le monde entier sur Amazon CloudFront. Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé.

Lorsque vous utilisez Transfer Acceleration, des frais supplémentaires de transfert de données peuvent s'appliquer. Pour plus d'informations sur la tarification, consultez Tarification Amazon S3.

Pourquoi utiliser Transfer Acceleration ?

Vous pouvez utiliser Transfer Acceleration sur un compartiment pour des raisons diverses :

- Vos clients effectuent les chargements vers un compartiment centralisé à partir du monde entier.
- Vous transférez régulièrement de plusieurs gigaoctets à plusieurs téraoctets de données d'un continent à l'autre.
- Vous ne pouvez pas utiliser toute la bande passante disponible sur Internet lors du chargement vers Amazon S3.

Pour plus d'informations sur le moment où utiliser Transfer Acceleration, consultez les FAQ Amazon S3.

Conditions d'utilisation de Transfer Acceleration

Les conditions suivantes sont requises pour utiliser Transfer Acceleration sur un compartiment S3 :

- Transfer Acceleration n'est prise en charge que sur les demandes de type hébergement virtuel. Pour plus d'informations sur les demandes de type hébergement virtuel, consultez Demandes à l'aide de l'API REST.
- Le nom du compartiment utilisé pour Transfer Acceleration doit être conforme aux règles DNS et ne doit pas inclure de points (« . »).

- Transfer Acceleration doit être activée sur le compartiment. Pour plus d'informations, consultez Activation et utilisation de S3 Transfer Acceleration.

  Une fois que vous activez Transfer Acceleration sur un compartiment, 20 minutes peuvent être nécessaires avant l'augmentation de la vitesse du transfert des données vers le compartiment.

  > **Note**
  >
  > Transfer Acceleration est actuellement prise en charge pour les compartiments situés dans les régions suivantes :
  >
  > - Asie-Pacifique (Tokyo) (ap-northeast-1)
  > - Asie-Pacifique (Séoul) (ap-northeast-2)
  > - Asie-Pacifique (Mumbai) (ap-south-1)
  > - Asie-Pacifique (Singapour) (ap-southeast-1)
  > - Asie-Pacifique (Sydney) (ap-southeast-2)
  > - Canada (Centre) (ca-central-1)
  > - Europe (Francfort) (eu-central-1)
  > - Europe (Irlande) (eu-west-1)
  > - Europe (Londres) (eu-west-2)
  > - Europe (Paris) (eu-west-3)
  > - Amérique du Sud (São Paulo) (sa-east-1)
  > - USA Est (Virginie du Nord) (us-east-1)
  > - USA Est (Ohio) (us-east-2)
  > - US Ouest (N. California) (us-west-1)
  > - USA Ouest (Oregon) (us-west-2)

- Pour accéder au compartiment activé pour Transfer Acceleration, vous devez utiliser le point de terminaison *`bucketname`*`.s3-accelerate.amazonaws.com`. Vous pouvez aussi utiliser le point de terminaison double pile *`bucketname`*`.s3-accelerate.dualstack.amazonaws.com` pour vous connecter au compartiment activé via IPv6. Vous pouvez continuer à utiliser les points de terminaison habituels pour le transfert de données standard.
- Vous devez être le propriétaire du compartiment pour définir l'état de l'accélération du transfert. Le propriétaire du compartiment peut attribuer des autorisations à d'autres utilisateurs pour leur permettre de définir l'état de l'accélération dans un compartiment. L'autorisation

`s3:PutAccelerateConfiguration` permet aux utilisateurs d'activer ou de désactiver Transfer Acceleration sur un compartiment. L'`s3:GetAccelerateConfiguration`autorisation permet aux utilisateurs de renvoyer l'état d'accélération du transfert d'un bucket, qui est `Enabled` soit `Suspended`.

Les sections suivantes décrivent la manière de mettre en route et utiliser Amazon S3 Transfer Acceleration pour transférer des données.

Rubriques

- Mise en route d'Amazon S3 Transfer Acceleration
- Activation et utilisation de S3 Transfer Acceleration
- Utilisation de l'outil de comparaison de vitesse Amazon S3 Transfer Acceleration

Mise en route d'Amazon S3 Transfer Acceleration

Vous pouvez utiliser Amazon S3 Transfer Acceleration pour obtenir un transfert rapide, facile et sécurisé des fichiers sur des longues distances entre votre client et un compartiment S3. Transfer Acceleration utilise les emplacements périphériques répartis dans le monde entier sur Amazon CloudFront. Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé.

Pour commencer à utiliser Amazon S3 Transfer Acceleration, effectuez les étapes suivantes :

1. Activer Transfer Acceleration sur un compartiment

   Vous pouvez activer Transfer Acceleration sur un compartiment de l'une des manières suivantes :

   - Utilisez la console Amazon S3.
   - Utilisez l'opération PUT Bucket accelerate de l'API REST.
   - Utilisez les AWS SDK AWS CLI et. Pour plus d’informations, consultez Développement avec Amazon S3 à l'aide des AWS SDK.

   Pour plus d’informations, consultez Activation et utilisation de S3 Transfer Acceleration.

> **Note**
>
> Pour que votre compartiment fonctionne avec l'accélération du transfert, son nom doit être conforme aux exigences d'attribution de noms DNS et ne doit pas contenir des points (« . »).

2. Transférer des données vers et depuis le compartiment activé pour l'accélération

   Utilisez l'un des noms de domaine des points de terminaison s3-accelerate suivants :

   - Pour accéder à un compartiment activé pour l'accélération, veuillez utiliser *bucketname*`.s3-accelerate.amazonaws.com`.
   - Pour accéder à un compartiment activé pour l'accélération via IPv6, veuillez utiliser *bucketname*`.s3-accelerate.dualstack.amazonaws.com`.

     Les points de terminaison à double pile Amazon S3 prennent en charge les demandes envoyées aux compartiments S3 via IPv6 et IPv4. Seul le point de terminaison double pile (« Dual-Stack ») Transfer Acceleration utilise le nom de point de terminaison de type hébergement virtuel. Pour plus d'informations, consultez Mise en route de l'envoi des demandes via IPv6 et Utilisation des points de terminaison Dual-Stack Amazon S3.

   > **Note**
   >
   > Votre application de transfert de données doit utiliser l'un des deux types de points de terminaison suivants pour accéder au compartiment afin d'accélérer le transfert de données : `.s3-accelerate.amazonaws.com` ou `.s3-accelerate.dualstack.amazonaws.com` pour le point de terminaison à double pile. Si vous souhaitez utiliser le transfert de données standard, vous pouvez continuer à utiliser les points de terminaison habituels.

Vous pouvez pointer vos demandes d'objet PUT et GET Amazon S3 sur le nom de domaine du point de terminaison `s3-accelerate` une fois que vous avez activé Transfer Acceleration. Par exemple, supposons que vous disposez actuellement d'une application API REST utilisant PUT Object qui utilise le nom d'hôte `mybucket.s3.us-east-1.amazonaws.com` dans la demande PUT. Pour accélérer le PUT, vous modifiez le nom d'hôte dans votre demande à `mybucket.s3-`

`accelerate.amazonaws.com`. Pour revenir à l'utilisation de la vitesse de chargement standard, remplacez le nom par `mybucket.s3.us-east-1.amazonaws.com`.

Une fois que Transfer Acceleration est activé, il peut s'écouler jusqu'à 20 minutes pour que vous preniez conscience de l'avantage offert en termes de performance. Cependant, le point de terminaison d'accélération est disponible dès que vous activez Transfer Acceleration.

Vous pouvez utiliser le point de terminaison Accelerate dans AWS les AWS CLI SDK et autres outils qui transfèrent des données vers et depuis Amazon S3. Si vous utilisez les AWS SDK, certains des langages pris en charge utilisent un indicateur de configuration du client de point de terminaison accéléré. Vous n'avez donc pas besoin de définir explicitement le point de terminaison sur lequel Transfer Acceleration est activé. *`bucketname`*`.s3-accelerate.amazonaws.com` Pour obtenir des exemples d'utilisation d'un indicateur de configuration client du point de terminaison d'accélération, consultez la pageActivation et utilisation de S3 Transfer Acceleration.

Vous pouvez utiliser toutes les opérations Amazon S3 via les points de terminaison d'accélération de transfert, sauf pour les éléments suivants :

- GET Service (répertorier les compartiments)
- PUT Bucket (créer un compartiment)
- DELETE Bucket

En outre, Amazon S3 Transfer Acceleration ne supporte pas les copies inter-Régions à l'aide de PUT Object - Copy.

Activation et utilisation de S3 Transfer Acceleration

Vous pouvez utiliser les fichiers de transfert Amazon S3 Transfer Acceleration rapidement et en toute sécurité sur de longues distances entre votre client et un compartiment S3. Vous pouvez activer l'accélération des transferts à l'aide de la console S3, du AWS Command Line Interface (AWS CLI), de l'API ou AWS des SDK.

Cette section propose des exemples d'activation d'Amazon S3 Transfer Acceleration sur un compartiment et d'utilisation du point de terminaison d'accélération pour le compartiment activé.

Pour en savoir plus sur les exigences de Transfer Acceleration, veuillez consulter Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Utiliser la console S3.

> **Note**
>
> Si vous souhaitez comparer les vitesses de chargement accéléré et non accéléré, ouvrez l'Outil de comparaison de la vitesse d'Amazon S3 Transfer Acceleration.
> L'outil de comparaison de vitesse utilise le téléchargement en plusieurs parties pour transférer un fichier de votre navigateur vers différents navigateurs Régions AWS avec et sans accélération de transfert Amazon S3. Vous pouvez comparer la vitesse de téléchargement des téléchargements directs et transférer des téléchargements accélérés par Région.

Pour activer Transfer Acceleration pour un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom du compartiment pour lequel vous souhaitez activer Transfer Acceleration.
3. Choisissez Propriétés.
4. Sous Transfer acceleration (Accélération du transfert), choisissez Edit (Modifier).
5. Choisissez Enable (Activer), puis Save changes (Enregistrer les modifications).

Pour accéder à des transferts de données accélérés

1. Une fois qu'Amazon S3 active l'accélération du transfert pour votre compartiment, affichez l'onglet Propriétés pour le compartiment.
2. Sous Transfer acceleration (Accélération du transfert), Accelerated endpoint (Point de terminaison accéléré) affiche le point de terminaison Transfer Acceleration pour votre compartiment. Utilisez ce point de terminaison pour accéder à des transferts de données accélérés vers et depuis votre compartiment.

   Si vous interrompez Transfer Acceleration, le point de terminaison de l'accélération ne fonctionne plus.

À l'aide du AWS CLI

Voici des exemples de AWS CLI commandes utilisées pour l'accélération des transferts. Pour obtenir des instructions sur la configuration du AWS CLI, voirDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

Activation de Transfer Acceleration sur un compartiment

Utilisez la AWS CLI put-bucket-accelerate-configurationcommande pour activer ou suspendre l'accélération du transfert sur un bucket.

L'exemple suivant définit `Status=Enabled` pour activer Transfer Acceleration sur un compartiment. Vous utilisez `Status=Suspended` pour suspendre Transfer Acceleration.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-
configuration Status=Enabled
```

Utilisation de Transfer Acceleration

Vous pouvez diriger toutes les demandes Amazon S3 effectuées par les AWS CLI commandes s3 et s3api vers le point de terminaison accéléré :. `s3-accelerate.amazonaws.com` Pour ce faire, définissez la valeur de configuration `use_accelerate_endpoint` sur `true` dans un profil de votre fichier AWS Config. Transfer Acceleration doit être activé sur votre compartiment si vous souhaitez utiliser le point de terminaison d'accélération.

Toutes les demandes sont envoyées en utilisant le style virtuel de l'adressage de compartiment : `my-bucket.s3-accelerate.amazonaws.com`. Les requêtes `ListBuckets`, `CreateBucket` et `DeleteBucket` ne sont pas envoyées au point de terminaison d'accélération, car celui-ci ne prend pas en charge ces opérations.

Pour plus d'informations sur `use_accelerate_endpoint`, consultez Configuration S3 d'AWS CLI dans la Référence des commandes AWS CLI .

L'exemple suivant définit `use_accelerate_endpoint` sur `true` dans le profil par défaut.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Si vous souhaitez utiliser le point de terminaison d'accélération pour certaines AWS CLI commandes mais pas pour d'autres, vous pouvez utiliser l'une des deux méthodes suivantes :

- Utilisez le point de terminaison d'accélération pour toutes les commandes s3 ou s3api en définissant le paramètre `--endpoint-url` sur `https://s3-accelerate.amazonaws.com`.
- Configurez des profils distincts dans votre fichier AWS Config. Par exemple, vous pouvez créer un profil qui définit `use_accelerate_endpoint` sur `true` et un profil qui ne définit pas `use_accelerate_endpoint`. Lorsque vous exécutez une commande, spécifiez le profil que vous souhaitez utiliser en fonction de votre intention d'utiliser le point de terminaison d'accélération.

Chargement d'un objet dans un compartiment activé pour Transfer Acceleration

L'exemple suivant charge un fichier sur un compartiment activé pour Transfer Acceleration en utilisant le profil par défaut qui a été configuré pour utiliser le point de terminaison d'accélération.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

L'exemple suivant charge un fichier sur un compartiment activé pour Transfer Acceleration en utilisant le paramètre `--endpoint-url` pour spécifier le point de terminaison d'accélération.

Example

```
$ aws configure set s3.addressing_style virtual
$ aws s3 cp file.txt s3://bucketname/keyname --region region --endpoint-url https://s3-accelerate.amazonaws.com
```

Utilisation des AWS SDK

Voici des exemples d'utilisation de Transfer Acceleration pour charger des objets sur Amazon S3 à l'aide du AWS SDK. Certains langages pris en charge par le AWS SDK (par exemple, Java et .NET) utilisent un indicateur de configuration du client de point de terminaison accéléré. Il n'est donc pas nécessaire de définir explicitement le point de terminaison pour Transfer Acceleration sur *bucketname* .s3-accelerate.amazonaws.com.

Java

Example

L'exemple suivant montre comment utiliser un point de terminaison d'accélération pour charger un objet sur Amazon S3. Cet exemple effectue les opérations suivantes :

- Crée un `AmazonS3Client` configuré pour utiliser les points de terminaison d'accélération. Tous les compartiments auxquels le client accède doivent avoir la fonction Transfer Acceleration activée.
- Active la fonction Transfer Acceleration sur un compartiment spécifié. Cette étape n'est nécessaire que si le compartiment que vous spécifiez ne possède pas déjà la fonction Transfer Acceleration activée.
- Vérifie que la fonction Transfer Acceleration est activée pour le compartiment spécifié.
- Charge un nouvel objet sur le compartiment spécifié à l'aide du point de terminaison d'accélération du compartiment.

Pour plus d'informations sur l'utilisation de Transfer Acceleration, consultez Mise en route d'Amazon S3 Transfer Acceleration. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

public class TransferAcceleration {
    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";

        try {
```

```
            // Create an Amazon S3 client that is configured to use the accelerate
 endpoint.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .enableAccelerateMode()
                    .build();

            // Enable Transfer Acceleration for the specified bucket.
            s3Client.setBucketAccelerateConfiguration(
                    new SetBucketAccelerateConfigurationRequest(bucketName,
                            new BucketAccelerateConfiguration(
                                    BucketAccelerateStatus.Enabled)));

            // Verify that transfer acceleration is enabled for the bucket.
            String accelerateStatus = s3Client.getBucketAccelerateConfiguration(
                    new GetBucketAccelerateConfigurationRequest(bucketName))
                    .getStatus();
            System.out.println("Bucket accelerate status: " + accelerateStatus);

            // Upload a new object using the accelerate endpoint.
            s3Client.putObject(bucketName, keyName, "Test object for transfer
 acceleration");
            System.out.println("Object \"" + keyName + "\" uploaded with transfer
 acceleration.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

L'exemple suivant montre comment utiliser le AWS SDK for .NET pour activer l'accélération du transfert sur un bucket. Pour plus d'informations sur la configuration et l'exécution des

exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TransferAccelerationTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            EnableAccelerationAsync().Wait();
        }

        static async Task EnableAccelerationAsync()
        {
                try
                {
                    var putRequest = new PutBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName,
                        AccelerateConfiguration = new AccelerateConfiguration
                        {
                            Status = BucketAccelerateStatus.Enabled
                        }
                    };
                    await
 s3Client.PutBucketAccelerateConfigurationAsync(putRequest);

                    var getRequest = new GetBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName
```

```
                    };
                    var response = await
 s3Client.GetBucketAccelerateConfigurationAsync(getRequest);

                    Console.WriteLine("Acceleration state = '{0}' ",
 response.Status);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    Console.WriteLine(
                        "Error occurred. Message:'{0}' when setting transfer
 acceleration",
                        amazonS3Exception.Message);
                }
        }
    }
}
```

Lors du chargement d'un objet sur un compartiment pour lequel Transfer Acceleration est activé, vous spécifiez l'utilisation du point de terminaison d'accélération au moment de la création d'un client.

```
var client = new AmazonS3Client(new AmazonS3Config
            {
                RegionEndpoint = TestRegionEndpoint,
                UseAccelerateEndpoint = true
            }
```

Javascript

Pour un exemple d'activation de l'accélération du transfert à l'aide du AWS SDK pour JavaScript, consultez la section Appel de l'opération putBucketAccelerate de configuration dans le AWS SDK pour la référence des JavaScript API.

Python (Boto)

Pour obtenir un exemple d'activation de Transfer Acceleration à l'aide du kit SDK pour Python, veuillez consulter put_bucket_accelerate_configuration dans la Référence API SDK pour Python (Boto3).

Other

Pour plus d'informations sur l'utilisation d'autres AWS SDK, voir Exemples de code et bibliothèques.

Utilisation de l'API REST

Utilisez l'opération `PutBucketAccelerateConfiguration` de l'API REST pour accélérer la configuration sur un compartiment existant.

Pour plus d'informations, consultez PutBucketAccelerateConfigurationle manuel Amazon Simple Storage Service API Reference.

Utilisation de l'outil de comparaison de vitesse Amazon S3 Transfer Acceleration

Vous pouvez utiliser l'outil de comparaison de vitesse Amazon S3 Transfer Acceleration pour comparer les vitesses de chargement accéléré et non accéléré dans les Régions Amazon S3. L'outil de comparaison de vitesse utilise les chargements partitionnés pour transférer un fichier à partir de votre navigateur vers différentes Régions Amazon S3 avec ou sans Transfer Acceleration.

Vous pouvez accéder à l'outil de comparaison de la vitesse à l'aide de l'une des méthodes suivantes :

- Copiez l'URL suivante dans la fenêtre de votre navigateur, en remplaçant *`region`* par Région AWS celle que vous utilisez (par exemple,`us-west-2`) et *`yourBucketName`*par le nom du bucket que vous souhaitez évaluer :

  ```
  https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/
  accelerate-speed-comparsion.html?
  region=region&origBucketName=yourBucketName
  ```

  Pour obtenir la liste des régions prises en charge par Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans la Références générales AWS.
- Utilisez la console Amazon S3.

Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation

C'est en général le propriétaire du compartiment qui prend en charge l'ensemble des frais de stockage et de transfert de données Amazon S3 associés à son compartiment. Toutefois, vous pouvez configurer un compartiment pour être un compartiment de type Paiement par le demandeur . Lorsqu'un compartiment est de type Paiement par le demandeur, les frais associés à la demande et au téléchargement de données depuis le compartiment sont facturés au demandeur, et non plus propriétaire du compartiment. En revanche, les frais de stockage des données sont toujours facturés au propriétaire du compartiment.

En général, nous recommandons de configurer des compartiments de type Paiement par le demandeur lorsque vous souhaitez partager des données, sans supporter les frais d'accès des autres utilisateurs. Par exemple, vous pouvez utiliser des compartiments Paiement par le demandeur lorsque vous proposez des ensembles de données volumineux, tels que des répertoires de codes postaux, des données de référence, des informations géospatiales ou des données d'indexation de site Web.

> **Important**
>
> Si vous activez la fonctionnalité Paiement par le demandeur, l'accès anonyme au compartiment n'est plus autorisé.

Vous devez authentifier toutes les demandes associées à vos compartiments de type Paiement par le demandeur. L'authentification des demandes permet à Amazon S3 d'identifier le demandeur et de lui facturer l'utilisation du compartiment de type Paiement par le demandeur.

Lorsque le demandeur assume un rôle AWS Identity and Access Management (IAM) avant de faire sa demande, le compte auquel appartient le rôle est débité pour la demande. Pour plus d'informations sur les rôles IAM, consultez Rôles IAM dans le manuel IAM Guide de l'utilisateur.

Une fois que vous avez configuré un bucket pour qu'il soit un bucket Requester Pays, les demandeurs doivent montrer qu'ils comprennent que la demande et le téléchargement des données leur seront facturés. Pour montrer qu'ils acceptent les frais, les demandeurs doivent soit inclure `x-amz-request-payer` comme en-tête dans leur demande d'API les requêtes DELETE, GET, HEAD, POST et PUT, soit ajouter le `RequestPayer` paramètre dans leur requête REST. Pour les demandes CLI, les demandeurs peuvent utiliser le `--request-payer` paramètre.

Example — Utiliser Requester Pays lors de la suppression d'un objet

Pour utiliser l'exemple DeleteObjectVersiond'API suivant, remplacez le *user input placeholders* par vos propres informations.

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
x-amz-expected-bucket-owner: ExpectedBucketOwner
```

Si le demandeur restaure des objets à l'aide de l'RestoreObjectAPI, Requester Pays est pris en charge tant que l'`x-amz-request-payer`en-tête ou le `RequestPayer` paramètre figurent dans la demande ; toutefois, le demandeur ne paie que le coût de la demande. Le propriétaire du bucket paie les frais de récupération.

Les compartiments de type Paiement par le demandeur sont incompatibles avec ce qui suit.

- Les demandes anonymes
- Requêtes SOAP
- Utiliser un compartiment de paiement par le demandeur comme compartiment cible pour la journalisation de l'utilisateur final, ou vice versa. Toutefois, vous pouvez activer la journalisation de l'utilisateur final sur un compartiment de type Paiement par le demandeur lorsque le compartiment cible n'est pas un compartiment de type Paiement par le demandeur.

Fonctionnement du Paiement par le demandeur

La facturation des demandes de type Paiement par le demandeur est très simple : la demande et le transfert de données sont facturés au demandeur et le stockage des données est facturé au propriétaire du compartiment. Cependant, la demande est facturée au propriétaire du compartiment dans les cas suivants :

- La demande renvoie une erreur `AccessDenied` (HTTP`403 Forbidden`) et la demande est initiée au sein du AWS compte individuel ou de l' AWS organisation du propriétaire du bucket.
- si la demande est une demande SOAP.

Pour plus d'informations sur le paiement par le demandeur, consultez les rubriques suivantes.

Rubriques

Configuration de Paiement par le demandeur sur un compartiment

Vous pouvez configurer un compartiment Amazon S3 comme compartiment de type Paiement par le demandeur afin que le demandeur paie à la place du propriétaire du compartiment le coût des demandes et des téléchargements des données.

Cette section fournit des exemples de configuration du Paiement par le demandeur sur un compartiment Amazon S3 à l'aide de la console et de l'API REST.

Utiliser la console S3.

Pour activer le paiement par le demandeur pour un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments) choisissez le nom du compartiment pour lequel vous souhaitez activer le paiement par le demandeur.
3. Choisissez Propriétés.
4. Sous Requester pays (Paiement par le demandeur), choisissez Edit (Modifier).
5. Choisissez Enable (Activer), puis Save changes (Enregistrer les modifications).

   Amazon S3 active le paiement par le demandeur pour le compartiment et affiche Bucket overview (Présentation du compartiment). Vous trouverez sous Paiement par le demandeur la mention Activé.

Utilisation de l'API REST

Seul le propriétaire du compartiment peut définir sa valeur de configuration `RequestPaymentConfiguration.payer` sur `BucketOwner` (valeur par défaut) ou sur

`Requester`. La configuration de la ressource `requestPayment` est facultative. Par défaut, le compartiment n'est pas de type Paiement par le demandeur.

Pour désactiver la fonctionnalité Paiement par le demandeur et revenir à un compartiment standard, utilisez la valeur `BucketOwner`. En général, la valeur `BucketOwner` est utilisée lors du chargement des données dans le compartiment Amazon S3, et la valeur `Requester` est ensuite définie avant la publication des objets dans le compartiment.

Pour configurer requestPayment

- Utilisez une demande PUT pour définir la valeur `Payer` sur `Requester` dans un compartiment donné.

```
PUT ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Si la demande s'exécute correctement, Amazon S3 renvoie une réponse similaire à la suivante :

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
x-amz-request-charged:requester
```

Vous pouvez définir le Paiement par le demandeur uniquement au niveau du regroupement. Vous ne pouvez pas définir le Paiement par le demandeur pour des objets spécifiques dans le compartiment.

Vous pouvez configurer un compartiment sur `BucketOwner` ou sur `Requester` à tout moment. Cependant, il peut y avoir quelques minutes avant que la nouvelle valeur de configuration ne prenne effet.

> **Note**
>
> Si vous êtes propriétaire d'un compartiment et que vous utilisez des URL pré-signées, nous vous recommandons de bien réfléchir avant de configurer votre compartiment pour en faire un compartiment de type Paiement par le demandeur (notamment si la durée de vie de l'URL est longue). En effet, le propriétaire du compartiment est facturé chaque fois qu'un demandeur utilise une URL pré-signée qui est associée aux informations d'identification du propriétaire du compartiment.

Récupération de la configuration requestPayment à l'aide de l'API REST

Vous pouvez demander la ressource `Payer` afin de connaître la valeur `requestPayment` définie pour un compartiment.

Pour obtenir la ressource requestPayment

- Utilisez une demande GET pour obtenir la ressource `requestPayment`, comme illustré dans la demande ci-dessous.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Si la demande s'exécute correctement, Amazon S3 renvoie une réponse similaire à la suivante :

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Type: [type]
Content-Length: [length]
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
```

```
</RequestPaymentConfiguration>
```

Cette réponse montre que la valeur `payer` est définie sur `Requester`.

Téléchargement d'objets depuis les compartiments Requester Pays

Le téléchargement de données depuis les compartiments de type Paiement par le demandeur étant facturé aux demandeurs, les demandes doivent contenir un paramètre spécifique, `x-amz-request-payer`, qui confirme que les demandeurs savent que le téléchargement leur sera facturé. Pour accéder à des objets stockés dans un compartiment de type Paiement par le demandeur, les demandeurs doivent intégrer à leurs demandes l'un des paramètres suivants.

- Les demandes DELETE, GET, HEAD, POST et PUT doivent intégrer `x-amz-request-payer : requester` dans l'en-tête.
- Les demandes associées à des URL signées doivent intégrer `x-amz-request-payer=requester`.

Lorsque la demande s'exécute correctement et que les frais sont facturés au demandeur, la réponse comprend l'en-tête `x-amz-request-charged:requester`. Si le paramètre `x-amz-request-payer` n'est pas inclus dans la demande, Amazon S3 renvoie une erreur 403 et la demande est facturée au propriétaire du compartiment.

> **Note**
>
> Il est inutile d'intégrer le paramètre `x-amz-request-payer` à vos demandes si vous être propriétaire du compartiment.
> En revanche, assurez-vous d'avoir intégré le paramètre `x-amz-request-payer` et la valeur qui lui est associée dans le calcul de la signature. Pour plus d'informations, consultez la section Construction de l' CanonicalizedAmzHeaders élément.

Utilisation de l'API REST

Pour télécharger des objets depuis un compartiment de type Paiement par le demandeur

- Utilisez une demande GET pour télécharger un objet depuis un compartiment de type Paiement par le demandeur, comme illustré dans la demande ci-dessous.

```
GET / [destinationObject] HTTP/1.1
Host: [BucketName].s3.amazonaws.com
x-amz-request-payer : requester
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Lorsque la demande GET s'exécute correctement et que les frais sont facturés au demandeur, la réponse comprend `x-amz-request-charged:requester`.

Lorsque les demandes tentent d'obtenir un objet à partir d'un compartiment de type Paiement par le demandeur, il peut arriver qu'Amazon S3 renvoie une erreur `Access Denied`. Pour plus d'informations, consultez Réponses d'erreur dans la Référence d'API Amazon Simple Storage Service.

À l'aide du AWS CLI

Pour télécharger des objets depuis un bucket Requester Pays à l'aide du AWS CLI, vous le spécifiez dans `--request-payer requester` le cadre de votre `get-object` demande. Pour plus d'informations, consultez get-object dans la Référence des commandes AWS CLI .

Limites et restrictions applicables aux compartiments

Un compartiment Amazon S3 appartient à Compte AWS celui qui l'a créé. La propriété du compartiment n'est pas transférable à un autre compte.

Lorsque vous créez un bucket, vous choisissez son nom et le Région AWS nom dans lequel le créer. Une fois le compartiment créé, vous ne pouvez pas changer son nom ni sa région.

Lorsque vous nommez un compartiment, choisissez un nom qui est pertinent pour vous ou votre entreprise. Évitez d'utiliser des noms associés à d'autres. Par exemple, vous devriez éviter d'utiliser `AWS` ou `Amazon` dans votre nom de compartiment.

Par défaut, vous pouvez créer jusqu'à 100 compartiments dans chacun de vos Comptes AWS. Si vous avez besoin de compartiments supplémentaires, vous pouvez augmenter votre quota de compartiments de compte à un maximum de 1 000 compartiments en soumettant une demande d'augmentation de quota. Il n'y a pas de différence au niveau des performances, que vous utilisiez de nombreux compartiments ou seulement quelques-uns.

> **Note**
>
> Il n'est pas nécessaire de soumettre plusieurs demandes d'augmentation de quota pour chacune d'entre elles Région AWS. Votre quota de compartiments est appliqué à votre Compte AWS.

Pour obtenir des informations sur la manière d'augmenter votre quota de compartiments, consultez Quotas de services AWS dans la Référence générale AWS .

Réutilisation des noms de compartiments

Si un compartiment est vide, vous pouvez le supprimer. Une fois qu'un compartiment a été supprimé, son nom peut être réutilisé. Toutefois, après avoir supprimé le compartiment, il se peut que vous ne puissiez pas réutiliser le nom pour diverses raisons.

Par exemple, lorsque vous supprimez le compartiment et que le nom est de nouveau disponible pour être réutilisé, un autre Compte AWS peut créer un compartiment avec ce nom. En outre, un certain temps peut s'écouler avant que vous ne puissiez réutiliser le nom d'un compartiment supprimé. Si vous souhaitez utiliser le même nom de compartiment, nous vous recommandons de ne pas supprimer le compartiment.

Pour plus d'informations sur les noms de compartiment, consultez Règles de dénomination de compartiment.

Limitations sur les objets et compartiments

Il n'y a pas de taille de compartiment maximale ou de limite au nombre d'objets que vous pouvez stocker dans un compartiment. Vous pouvez également choisir de stocker tous vos objets dans un seul compartiment ou les répartir dans différents compartiments. Toutefois, vous ne pouvez pas créer un compartiment au sein d'un autre compartiment.

Opérations de compartiment

L'ingénierie haute disponibilité d'Amazon S3 met l'accent sur les opérations get, put, list et delete. Étant donné que les opérations associées aux compartiments fonctionnent dans un espace de ressources mondial et centralisé, il n'est pas recommandé de créer, supprimer ni configurer de compartiments dans le chemin de code haute disponibilité de votre application. Il est préférable de créer, supprimer ou configurer des compartiments dans le cadre de routines d'initialisation ou de configuration distinctes, que vous exécutez moins souvent.

Attribution des noms de compartiments et compartiments créés automatiquement

Si votre application crée des compartiment automatiquement, pensez à choisir un schéma d'attribution de noms qui ne soit pas susceptible d'entraîner des conflits de nom. Vous devez veiller à ce que la logique applicative sélectionne un autre nom lorsqu'un nom de compartiment est déjà utilisé.

Pour plus d'informations sur l'attribution de noms à des compartiments, reportez-vous à la section Règles de dénomination de compartiment.

Chargement, téléchargement et utilisation des objets dans Amazon S3

Pour stocker vos données dans Amazon S3, vous devez utiliser avec des ressources appelées compartiments et objets. Un compartiment est un conteneur d'objets. Un objet est un fichier et toutes les métadonnées qui le décrivent.

Pour stocker un objet dans Amazon S3, vous devez créer un compartiment, puis télécharger l'objet dans un compartiment. Lorsque l'objet se trouve dans le compartiment, vous pouvez l'ouvrir, le télécharger et le copier. Lorsque vous n'avez plus besoin d'un objet ou d'un compartiment, vous pouvez nettoyer ces ressources.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

> **Important**
>
> Dans la console Amazon S3, lorsque vous choisissez Open (Ouvrir) ou Download As (Télécharger en tant que) pour un objet, ces opérations créent des URL pré-signées. Pendant cinq minutes, votre objet sera accessible à toute personne ayant accès à ces URL pré-signées. Pour plus d'informations sur les URL pré-signées, consultez Utilisation d'URL pré-signées.

Avec Amazon S3, vous ne payez que les services que vous utilisez. Pour plus d'informations sur les fonctionnalités et les tarifs d'Amazon S3, consultez Amazon S3. Si vous êtes un nouveau client Amazon S3, vous pouvez commencer à utiliser Amazon S3 gratuitement. Pour plus d'informations, consultez la page sur l'offer gratuite AWS.

Rubriques

- Utilisation des métadonnées d'objet
- Chargement d'objets
- Chargement et copie d'objets à l'aide d'un chargement partitionné
- Copier, déplacer et renommer des objets
- Téléchargement d'objets
- Vérification de l'intégrité des objets
- Suppression d'objets Amazon S3
- Organisation, liste et utilisation de vos objets
- Utilisation d'URL présignées
- Transformation d'objets avec S3 Object Lambda

Présentation des objets Amazon S3

Amazon S3 est un magasin d'objets qui utilise des valeurs clés uniques pour stocker autant d'objets que vous le souhaitez. Vous stockez ces objets dans un ou plusieurs compartiments, et chaque objet peut atteindre 5 To. Un objet se compose des éléments suivants :

Key

Nom que vous attribuez à un objet. Vous utilisez la clé de l'objet pour récupérer l'objet. Pour plus d'informations, consultez Utiliser les métadonnées d'un objet.

ID de version

Dans un compartiment, une clé et un ID de version identifient un objet de manière unique. L'ID de version est une chaîne de caractères générée par Amazon S3 lorsque vous ajoutez un objet à un compartiment. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Valeur

Contenu que vous stockez.

Une valeur d'objet peut être toute séquence d'octets. La taille des objets peut aller de zéro à 5 To. Pour plus d'informations, consultez Chargement d'objets.

Metadonnées

Ensemble de paires nom-valeur grâce auxquelles vous pouvez stocker des informations sur l'objet. Vous pouvez attribuer des métadonnées, appelées métadonnées définies par l'utilisateur, aux objets dans Amazon S3. Amazon S3 attribue également des métadonnées système à ces objets pour les gérer. Pour plus d'informations, consultez Utiliser les métadonnées d'un objet.

Sous-ressources

Amazon S3 utilise le mécanisme des sous-ressources pour stocker des informations supplémentaires propres à l'objet. Etant donné que les sous-ressources sont subordonnées aux objets, elles sont toujours associées à d'autres entités comme un objet ou un compartiment. Pour plus d’informations, consultez Sous-ressources d'objet.

Informations sur le contrôle d'accès

Vous pouvez contrôler l'accès aux objets que vous stockez dans Amazon S3. Amazon S3 prend en charge le contrôle d'accès basé sur les ressources, comme la liste de contrôle d'accès (ACL) et les stratégies de compartiment, ainsi que le contrôle d'accès basé sur l'utilisateur. Pour en savoir plus sur le contrôle d'accès, consultez la rubrique suivante :

- Gestion des accès
- Identity and Access Management pour Amazon S3
- Configuration des listes ACL

Par défaut, vos ressources Amazon S3 (par exemple, les compartiments et les objets) sont privées. Vous devez accorder explicitement l'autorisation pour que d'autres utilisateurs accèdent aux ressources. Pour en savoir plus sur le partage d'objets, consultez Partage d'objets à l'aide d'URL présignées.

Balises

Vous pouvez utiliser des balises pour classer vos objets stockés par catégories à des fins de contrôle d'accès ou de répartition des coûts. Pour plus d’informations, consultez Catégorisation de votre stockage à l'aide de balises.

Sous-ressources d'objet

Amazon S3 définit un ensemble de sous-ressources associées à des compartiments et à des objets. Les sous-ressources sont subordonnées aux objets. Cela signifie que les sous-ressources n'existent

pas par elles-mêmes. Elles sont toujours associées à une autre entité, comme un objet ou un compartiment.

Le tableau suivant répertorie les sous-ressources associées aux objets Amazon S3.

Sous-ressource	Description
liste acl	Contient une liste d'accords identifiant les bénéficiaires et les autorisations accordées. Lorsque vous créez un objet, la sous-ressource `acl` identifie le propriétaire de l'objet comme ayant le contrôle total sur l'objet. Vous pouvez récupérer la liste ACL d'un objet ou la remplacer par une liste mise à jour d'attributions. Toute mise à jour apportée à une liste ACL exige un remplacement de la liste ACL existante. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

Création de noms de clés d'objet

La clé d'objet (ou nom de clé) identifie de façon unique l'objet dans un compartiment Amazon S3. Les métadonnées d'objet sont un ensemble de paires de noms-valeurs. Pour plus d'informations sur les métadonnées d'objet, consultez Utilisation des métadonnées d'objet.

Lorsque vous créez un objet, vous spécifiez le nom de clé, qui identifie de façon unique l'objet dans le compartiment. Par exemple, dans la console Amazon S3, lorsque vous mettez en évidence un compartiment, la liste des objets du compartiment s'affiche. Ces noms sont les clés d'objet. Le nom de clé d'objet est une séquence de caractères Unicode codés en UTF-8 dont la longueur maximale est de 1 024 octets. Les noms de clés d'objet ne sont pas sensibles à la casse.

> **Note**
>
> Les noms de clé d'objet avec la valeur « soap » ne sont pas pris en charge pour les virtual-hosted-style requêtes. Pour les noms de clé d'objet contenant la valeur « soap », une URL de type chemin d'accès doit être utilisée à la place.

Le modèle de données Amazon S3 est une structure horizontale : vous créez un compartiment et ce compartiment stocke des objets. Il n'existe aucune hiérarchie de sous-compartiments ou de sous-

dossiers. Toutefois, vous pouvez déduire une hiérarchie logique grâce aux préfixes et délimiteurs de nom de clé à l'image de la console Amazon S3. La console Amazon S3 prend en charge un concept de dossiers. Pour en savoir plus sur la modification des métadonnées à partir de la console Amazon S3, veuillez consulter Modification des métadonnées d'objet dans la console Amazon S3.

Imaginons que le compartiment (`admin-created`) comporte quatre objets avec les clés d'objet suivantes :

`Development/Projects.xls`

`Finance/statement1.pdf`

`Private/taxdocument.pdf`

`s3-dg.pdf`

La console utilise les préfixes de nom de clé (`Development/`, `Finance/` et `Private/`) ainsi que le délimiteur (« / ») pour présenter une structure de dossiers. La clé `s3-dg.pdf` ne possède pas de préfixe, son objet apparaît donc directement à la racine du compartiment. Si vous ouvrez le dossier `Development/`, vous voyez qu'il contient l'objet `Projects.xlsx`.

- Amazon S3 prend en charge les compartiments et les objets. Il n'y a aucune hiérarchie. Cependant, en utilisant des préfixes et des délimiteurs dans le nom d'une clé d'objet, la console Amazon S3 et les AWS SDK peuvent déduire une hiérarchie et introduire le concept de dossiers.
- La console Amazon S3 implémente la création d'objets de dossier en créant un objet de type zéro octet avec la valeur préfixe et délimiteur du dossier comme clé. Ces objets de dossier n'apparaissent pas dans la console. Sinon, ils se comportent comme n'importe quel autre objet et peuvent être visualisés et manipulés via l'API REST, la AWS CLI et AWS les SDK.

Directives de dénomination de la clé d'objet

Vous pouvez utiliser n'importe quel caractère UTF-8 dans le nom de clé d'un objet. L'utilisation de certains caractères dans les noms de clé peut toutefois générer des problèmes avec certaines applications et certains protocoles. Les directives suivantes vous aident à optimiser la conformité avec le DNS, les -caractères adaptés pour le web, les analyseurs XML et les autres API.

Caractères adaptés

Les caractères configurés suivants sont généralement adaptés à une utilisation dans les noms de clés.

Caractères alphanumériques	• 0-9 • a-z • A-Z
Caractères spéciaux	• Point d'exclamation (!) • Trait d'union (-) • Trait de soulignement (_) • Point (.) • Astérisque (*) • Guillemet simple (') • Parenthèse ouvrante (() • Parenthèse fermante ())

Voici des exemples de noms de clés d'objet valides :

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Note**
>
> Pour les objets téléchargés à l'aide de la console Amazon S3, dont le nom de clé se termine par un ou plusieurs points « . », ceux-ci sont supprimés du nom de clé de l'objet téléchargé. Pour télécharger un objet dont le nom de clé se termine par un ou plusieurs points «. » conservé dans l'objet téléchargé, vous devez utiliser le AWS Command Line Interface (AWS CLI), AWS les SDK ou l'API REST.
>
> En outre, vous devez connaître les limitations suivantes en ce qui concerne les préfixes :
>
> - Les objets avec le préfixe « ./ » doivent être chargés ou téléchargés avec le AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST. Vous ne pouvez pas utiliser la console Amazon S3.
> - Les objets dont le préfixe est « ../ » ne peuvent pas être chargés à l'aide de la commande AWS Command Line Interface (AWS CLI) ou la console Amazon S3.

Caractères pouvant exiger une manipulation spéciale

Les caractères suivants dans un nom de clé peuvent exiger une manipulation de code supplémentaire et ont probablement besoin d'être encodés en URL ou référencés comme valeur HEX. Certains de ces caractères ne sont pas imprimables et le navigateur peut ne pas réussir à les traiter, ce qui exige également une manipulation spéciale :

- Esperluette (« & »)
- Dollar (« $ »)
- Les caractères ASCII 00–1F hex (0–31 décimale) et 7F (127 décimale)
- Arobase (« @ »)
- Egal (« = »)
- Point-virgule (« ; »)
- Barre oblique (« / »)
- Deux-points (« : »)
- Plus (« + »)
- Espace – Des séquences d'espaces significatives peuvent être perdues dans certaines utilisations (notamment les espaces multiples)
- Virgule (« , »)
- Point d'interrogation (« ? »)

Caractères à éviter

Nous vous recommandons de ne pas utiliser les caractères suivants dans le nom d'une clé en raison de la gestion importante des caractères spéciaux, qui n'est pas uniforme dans toutes les applications.

- Barre oblique inverse (« \ »)
- Accolade gauche (« { »)
- Caractères ASCII non imprimables (128–255 caractères décimaux)
- Lambda (« ^ »)
- Accolade droite (« } »)
- Pourcentage (« % »)
- Accent grave/guillemet inversé (« ` »)
- Crochet droit («] »)

- Guillemets
- Supérieur à (« > »)
- Crochet gauche (« [»)
- Tilde (« ~ »)
- Symbole « inférieur à » (« < »)
- Dièse (« # »)
- Barre verticale/pipe (« | »)

Contraintes de clé d'objet XML

Comme spécifié par la norme XML relative à la end-of-line manipulation, tout le texte XML est normalisé de telle sorte que les retours d'un seul chariot (code ASCII 13) et les retours de chariot immédiatement suivis d'un flux de ligne (code ASCII 10) sont remplacés par un caractère d'alimentation d'une seule ligne. Pour analyser correctement les clés d'objet dans les requêtes XML, les retours chariot et les autres caractères spéciaux doivent être remplacés par leur code d'entité XML équivalent lorsqu'ils sont insérés dans des balises XML. Voici la liste de ces caractères spéciaux et de leurs codes d'entité équivalents :

- ' comme `'`
- ” comme `"`
- & comme `&`
- < comme `<`
- > comme `>`
- \ r comme `
` ou `
`
- \ n comme `
` ou `
`

Example

L'exemple suivant montre l'utilisation d'un code d'entité XML à la place d'un retour chariot. Cette demande `DeleteObjects` supprime un objet avec le paramètre `key` : `/some/prefix/objectwith\rcarriagereturn` (où \ r est le retour chariot).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
```

```
  </Object>
</Delete>
```

Utilisation des métadonnées d'objet

Vous pouvez définir des métadonnées d'objet dans Amazon S3 au moment du chargement de l'objet. Les métadonnées d'objet sont un ensemble de paires de noms-valeurs. Une fois l'objet chargé, vous ne pouvez pas modifier ses métadonnées. Le seul moyen de modifier les métadonnées d'objet est de faire une copie de l'objet et de configurer les métadonnées.

Lorsque vous créez un objet, vous spécifiez également le nom de clé, qui identifie de façon unique l'objet dans le compartiment. La clé d'objet (ou nom de clé) identifie de façon unique l'objet dans un compartiment Amazon S3. Pour plus d'informations, consultez Création de noms de clés d'objet.

Il existe deux types de métadonnées dans Amazon S3 : les métadonnées définies par le système et les métadonnées définies par l'utilisateur. Les sections ci-dessous fournissent plus d'informations sur les métadonnées définies par le système et celles définies par l'utilisateur. Pour en savoir plus sur la modification des métadonnées à l'aide de la console Amazon S3, veuillez consulter Modification des métadonnées d'objet dans la console Amazon S3.

Métadonnées d'objet définies par le système

Pour chaque objet stocké dans un compartiment, Amazon S3 gère un ensemble de métadonnées système. Amazon S3 traite ces métadonnées système selon les besoins. Par exemple, Amazon S3 gère les métadonnées de date et de taille de création de l'objet et utilise ces informations dans le cadre de la gestion de l'objet.

Il existe deux classes de métadonnées système :

- Contrôlées par le système : les métadonnées comme la date de création de l'objet sont contrôlées par le système et seul Amazon S3 peut modifier la valeur.
- Contrôlées par l'utilisateur : les autres métadonnées système, comme la classe de stockage configurée pour l'objet et l'activation du chiffrement côté serveur dans l'objet, sont des exemples de métadonnées système dont vous contrôlez la valeur. Si le compartiment est configuré en tant que site web, il se peut que vous souhaitiez rediriger une demande de page vers une autre page ou une URL externe. Dans ce cas, une page Web est un objet dans le compartiment. Amazon S3 stocke la valeur de redirection de la page comme des métadonnées système dont vous contrôlez la valeur.

Lorsque vous créez des objets, vous pouvez configurer les valeurs de ces éléments de métadonnées système ou mettre à jour les valeurs le cas échéant. Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Amazon S3 utilise des AWS KMS clés pour chiffrer vos objets Amazon S3. AWS KMS chiffre uniquement les données de l'objet. Le total de contrôle, ainsi que l'algorithme spécifié, sont stockés dans le cadre des métadonnées de l'objet. Si un chiffrement côté serveur est demandé pour l'objet, le total de contrôle est stocké sous forme chiffrée. Pour plus d'informations sur le chiffrement côté serveur, consultez Protection des données à l'aide du chiffrement.

> **Note**
>
> L'en-tête de la demande PUT est limité à une taille de 8 Ko. Dans l'en-tête de la demande PUT, les métadonnées définies par le système sont limitées à une taille de 2 Ko. La taille des métadonnées définies par le système est mesurée grâce à la somme des octets dans l'encodage US-ASCII de chaque clé et valeur.

Le tableau suivant fournit une liste des métadonnées définies par le système et indique si elles peuvent être mises à jour.

Nom	Description	L'utilisateur peut-il modifier la valeur ?
`Date`	Les date et heure actuelles.	Non
`Cache-Control`	Champ d'en-tête général utilisé pour spécifier les stratégies de mise en cache.	Oui
`Content-Disposition`	Informations de présentation des objets.	Oui
`Content-Length`	La taille des objets en octets.	Non

Nom	Description	L'utilisateur peut-il modifier la valeur ?
`Content-Type`	Le type d'objet.	Oui
`Last-Modified`	La date de création ou de dernière modification, la plus récente des deux. Pour les chargements partitionnés, la date de création de l'objet correspond à la date de lancement du chargement partitionné.	Non
`ETag`	Une balise d'entité (ETag) représente une version spécifique de cet objet. Pour les objets qui ne sont pas chargés en tant que chargement partitionné et qui sont non chiffrés ou chiffrés par chiffrement côté serveur avec des clés gérées par Simple Storage Service (Amazon S3) (SSE-S3), l'ETag est une valeur de hachage MD5 des données.	Non
`x-amz-server-side-encryption`	En-tête qui indique si le chiffrement côté serveur est activé pour l'objet et si ce chiffrement utilise les clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou les clés de chiffrement gérées par Amazon S3 (SSE-S3). Pour plus d'informations, consultez Protection des données avec le chiffrement côté serveur.	Oui
`x-amz-checksum-crc32` , `x-amz-checksum-crc32c` , `x-amz-checksum-sha1` , `x-amz-checksum-sha256`	Des en-têtes qui contiennent le total de contrôle ou le récapitulatif de l'objet. Au maximum, un de ces en-têtes sera défini à la fois, en fonction de l'algorithme de total de contrôle que vous demandez à Amazon S3 d'utiliser. Pour plus d'informations sur la façon de sélectionner l'algorithme de total de contrôle,, consultez Vérification de l'intégrité des objets.	Non

Nom	Description	L'utilisateur peut-il modifier la valeur ?
`x-amz-version-id`	La version d'un objet. Lorsque vous activez la gestion des versions sur un compartiment, Amazon S3 attribue un ID de version aux objets ajoutés au compartiment. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.	Non
`x-amz-delete-marker`	Un marqueur booléen qui indique si l'objet est un marqueur de suppression. Ce marqueur n'est utilisé que dans les compartiments pour lesquels la gestion des versions est activée.	Non
`x-amz-storage-class`	La classe de stockage utilisée pour stocker l'objet. Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).	Oui
`x-amz-website-redirect-location`	Un en-tête qui redirige les demandes pour l'objet associé vers un autre objet dans le même compartiment ou une URL externe. Pour plus d'informations, consultez (Facultatif) Configuration de la redirection de pages web.	Oui
`x-amz-server-side-encryption-aws-kms-key-id`	En-tête qui indique l'ID de la clé KMS de chiffrement AWS KMS symétrique utilisée pour chiffrer l'objet. Cet en-tête est utilisé uniquement lorsque l'en-tête `x-amz-server-side-encryption` est présent et a la valeur de `aws:kms`.	Oui
`x-amz-server-side-encryption-customer-algorithm`	Un en-tête qui indique si le chiffrement côté serveur à l'aide des clés de chiffrement fournies par le client (SSE-C) est activé. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C).	Oui

Nom	Description	L'utilisateur peut-il modifier la valeur ?
`x-amz-tagging`	Ensemble de balises de l'objet. L'ensemble de balises doit être codé en tant que paramètres de requête URL.	Oui

Métadonnées d'objet définies par l'utilisateur

Lorsque vous chargez un objet, vous pouvez également lui attribuer des métadonnées. Vous fournissez ces informations facultatives en tant que paire nom-valeur (clé-valeur) lorsque vous envoyez une demande PUT ou POST pour créer l'objet. Lors du chargement des objets grâce à l'API REST, les noms facultatifs des métadonnées définies par l'utilisateur doivent commencer par « `x-amz-meta-` » pour se différencier des autres en-têtes HTTP. Lorsque vous récupérez l'objet grâce à l'API REST, ce préfixe est renvoyé. Lors du chargement des objets grâce à l'API SOAP, le préfixe n'est pas requis. Lorsque vous récupérez l'objet grâce à l'API SOAP, le préfixe est supprimé, quelle que soit l'API utilisée pour charger l'objet.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Au lieu d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Lorsque des métadonnées sont récupérées via l'API REST, Amazon S3 associe les en-têtes avec le même nom (sans tenir compte de la casse) dans une liste séparée par des virgules. Si certaines métadonnées contiennent des caractères non imprimables, elles ne sont pas renvoyées. À la place, l'en-tête `x-amz-missing-meta` est renvoyé avec le nombre d'entrées de métadonnées non imprimables. L'action `HeadObject` récupère les métadonnées d'un objet sans retourner l'objet lui-même. Cette opération est pratique si vous êtes uniquement intéressé par les métadonnées d'un objet. Pour utiliser `HEAD`, vous devez disposer d'un accès `READ` sur l'objet. Pour plus d'informations, consultez HeadObjectle manuel Amazon Simple Storage Service API Reference.

Les métadonnées définies par l'utilisateur sont un ensemble de paires de clés-valeurs. Amazon S3 stocke les clés de métadonnées définies par l'utilisateur en minuscules.

Amazon S3 accepte les caractères Unicode arbitraires dans vos valeurs de métadonnées.

Pour éviter des problèmes liés à la présentation de ces valeurs de métadonnées, vous devez vous conformer à l'utilisation de caractères US-ASCII avec REST et UTF-8 lors de l'utilisation de SOAP ou de chargements basés sur un navigateur via POST.

Lorsque vous utilisez des caractères non US-ASCII dans vos valeurs de métadonnées, la chaîne Unicode fournie est examinée pour les caractères non US-ASCII. Les valeurs de ces en-têtes sont décodées en caractères selon la RFC 2047 avant d'être stockées, puis encodées selon la RFC 2047 pour qu'elles soient transmises de manière sécurisée par e-mail avant d'être renvoyées. Si la chaîne contient uniquement des caractères US-ASCII, elle est présentée telle qu'elle est.

Voici un exemple.

```
PUT /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÕÑ S3

HEAD /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

> **Note**
>
> L'en-tête de la demande PUT est limité à une taille de 8 Ko. Dans l'en-tête de la demande PUT, les métadonnées définies par l'utilisateur sont limitées à une taille de 2 Ko. La taille des métadonnées définies par l'utilisateur est mesurée grâce à la somme des octets dans l'encodage UTF-8 de chaque clé et valeur.

Pour plus d'informations sur la modification des métadonnées de votre objet après son téléchargement, en créant une copie de l'objet, en le modifiant et en remplaçant l'ancien objet, ou en créant une version, consultez Modification des métadonnées d'objet dans la console Amazon S3.

Modification des métadonnées d'objet dans la console Amazon S3

Vous pouvez utiliser la console Amazon S3 pour modifier les métadonnées des objets S3 existants. Certaines métadonnées sont définies par Amazon S3 lorsque vous chargez l'objet. Par exemple, `Content-Length` et `Last-Modified` sont des champs de métadonnées d'objet définies par le système qui ne peuvent pas être modifiés par un utilisateur.

Vous pouvez également définir certaines métadonnées lorsque vous chargez l'objet et les ajouter ultérieurement en fonction de vos besoins. Par exemple, vous pouvez avoir un ensemble d'objets que vous stockez initialement dans la classe de stockage `STANDARD`. Au fil du temps, vous n'aurez peut-être plus besoin que ces données soient hautement disponibles. Vous modifiez donc la classe de stockage en `GLACIER` en modifiant la valeur de la clé `x-amz-storage-class` de `STANDARD` à `GLACIER`.

> **Note**
>
> Tenez compte des problèmes suivants lorsque vous modifiez des métadonnées d'objet dans Amazon S3 :
>
> - Cette action crée une copie de l'objet avec les paramètres mis à jour et la date de la dernière modification. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Si la gestion des versions S3 n'est pas activée, une nouvelle copie de l'objet remplace l'objet d'origine. Le rôle IAM Compte AWS associé qui modifie la propriété devient également propriétaire du nouvel objet ou (version de l'objet).
> - Pour utiliser la console Amazon S3 afin de modifier les métadonnées d'un objet doté de balises définies par l'utilisateur, vous devez également avoir l'`s3:GetObjectTagging`autorisation. Si vous utilisez la console Amazon S3 pour modifier les métadonnées d'un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également avoir l'`s3:GetObjectTagging`autorisation.

Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, les métadonnées de l'objet seront mises à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

- La modification des métadonnées met à jour les valeurs des noms de clés existants.
- Les objets chiffrés avec des clés de chiffrement fournies par le client (SSE-C) ne peuvent pas être copiés à l'aide de la console. Vous devez utiliser AWS CLI le AWS SDK ou l'API REST Amazon S3.

Warning

Lors de la modification des métadonnées de dossiers, attendez la fin de l'opération `Edit metadata` avant d'ajouter de nouveaux objets au dossier. Sinon, de nouveaux objets peuvent également être modifiés.

Les rubriques suivantes décrivent comment modifier les métadonnées d'un objet à l'aide de la console Amazon S3.

Modification de métadonnées définies par le système

Vous pouvez configurer certaines métadonnées système pour un objet S3, mais vous ne pouvez pas toutes les configurer. Pour obtenir la liste des métadonnées définies par le système et savoir si vous pouvez modifier leurs valeurs, veuillez consulter Métadonnées d'objet définies par le système.

Pour modifier les métadonnées définies par le système d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Accédez à votre compartiment ou dossier Amazon S3 et cochez la case située à gauche des noms des objets comportant des métadonnées que vous souhaitez modifier.
3. Dans le menu Actions, choisissez Modifier les actions, puis Modifier les métadonnées.
4. Passez en revue les objets répertoriés et choisissez Add metadata (Ajouter des métadonnées).
5. Pour Type de métadonnées, sélectionnez System-defined (Définies par le système).
6. Spécifiez une valeur Key (Clé) unique et renseignez le champ Value (Valeur) des métadonnées.

7. Pour modifier des métadonnées supplémentaires, choisissez Add metadata (Ajouter des métadonnées). Vous pouvez également choisir Supprimer pour supprimer un ensemble de type-key-values.
8. Lorsque vous avez terminé, choisissez Save changes (Enregistrer les modifications) pour qu'Amazon S3 modifie les métadonnées des objets spécifiés.

Modification de métadonnées définies par l'utilisateur

Vous pouvez modifier les métadonnées définies par l'utilisateur d'un objet en combinant le préfixe de métadonnées, `x-amz-meta-`, et un nom que vous choisissez pour créer une clé personnalisée. Par exemple, si vous ajoutez le nom personnalisé `alt-name`, la clé de métadonnées est `x-amz-meta-alt-name`.

Les métadonnées définies par l'utilisateur peuvent atteindre 2 Ko. Pour calculer la taille totale des métadonnées définies par l'utilisateur, additionnez le nombre d'octets dans l'encodage UTF-8 pour chaque clé et valeur. Les clés et leurs valeurs doivent respecter les normes US-ASCII. Pour plus d'informations, consultez Métadonnées d'objet définies par l'utilisateur.

Pour modifier les métadonnées définies par l'utilisateur d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom du compartiment qui contient les objets auxquels vous souhaitez ajouter des métadonnées.

   Vous pouvez également naviguer vers un dossier si vous le souhaitez.
3. Dans la liste Objets, cochez la case à côté des noms des objets auxquels vous souhaitez ajouter des métadonnées.
4. Dans le menu Actions, choisissez Modifier les métadonnées.
5. Passez en revue les objets répertoriés et choisissez Add metadata (Ajouter des métadonnées).
6. Pour le Type de métadonnées, choisissez Définies par l'utilisateur.
7. Renseignez le champ Cléavec une clé personnalisée unique au format `x-amz-meta-`. Renseignez également le champ Value (Valeur) pour les métadonnées.
8. Pour ajouter des métadonnées supplémentaires, choisissez Add metadata (Ajouter des métadonnées). Vous pouvez également choisir Supprimer pour supprimer un ensemble de type-key-values.

9. Choisissez Edit metadata (Modifier les métadonnées).

   Amazon S3 modifie les métadonnées des objets spécifiés.

Chargement d'objets

Lorsque vous chargez un fichier dans Amazon S3, il est stocké en tant qu'objet S3. Les objets se composent des données du fichier et des métadonnées décrivant l'objet. Vous pouvez disposer d'un nombre illimité d'objets dans un compartiment. Avant de pouvoir charger des fichiers dans un compartiment Amazon S3, vous devez disposer d'autorisations en écriture pour le compartiment. Pour plus d'informations sur les autorisations d'accès, consultez Identity and Access Management pour Amazon S3.

Vous pouvez charger n'importe quel type de fichier (images, sauvegardes, données, films, etc.) dans un compartiment S3. La console Amazon S3 vous permet de charger des fichiers d'une taille maximale de 160 Go. Pour charger un fichier de plus de 160 Go, utilisez le AWS Command Line Interface (AWS CLI), AWS les SDK ou l'API REST Amazon S3.

Si vous chargez un objet avec un nom de clé qui existe déjà dans un compartiment pour lequel le contrôle de version est activé, Amazon S3 crée une autre version de l'objet au lieu de remplacer l'objet existant. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la console S3.

Selon la taille des données chargées, Amazon S3 propose les options suivantes :

- Téléchargez un objet en une seule opération à l'aide AWS des SDK, de l'API REST ou AWS CLI : en une seule PUT opération, vous pouvez télécharger un seul objet d'une taille maximale de 5 Go.
- Charger un seul objet à l'aide de la console Amazon S3 : avec la console Amazon S3, vous pouvez charger un seul objet d'une taille maximale de 160 Go.
- Téléchargez un objet en plusieurs parties à l'aide AWS des SDK, de l'API REST ou AWS CLI : à l'aide de l'opération d'API de téléchargement en plusieurs parties, vous pouvez télécharger un seul objet volumineux d'une taille maximale de 5 To.

  L'opération d'API de chargement partitionné est conçue pour améliorer l'expérience de chargement pour les objets plus volumineux. Vous pouvez charger un objet en plusieurs parties. Ces parties d'objet peuvent être chargées indépendamment, dans n'importe quel ordre, et en parallèle. Vous pouvez utiliser un chargement partitionné pour les objets allant d'une taille maximale de 5 Mo à

5 To. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

Lorsque vous chargez un objet, il est automatiquement chiffré à l'aide du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) par défaut. Lorsque vous le téléchargez, l'objet est déchiffré. Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3 et Protection des données à l'aide du chiffrement.

Lorsque vous chargez un objet, si vous souhaitez utiliser un autre type de chiffrement par défaut, vous pouvez également spécifier le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) dans vos demandes PUT S3 ou définir la configuration de chiffrement par défaut dans le compartiment de destination afin d'utiliser SSE-KMS pour chiffrer vos données. Pour en savoir plus sur SSE-KMS, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS). Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez avoir l'autorisation d'utiliser la clé. Pour plus d'informations sur les autorisations intercomptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service .

Si vous rencontrez une erreur d'accès refusé (403 Interdit) dans Amazon S3, consultez Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3 pour en savoir plus sur ses causes courantes.

Utilisation de la console S3

Cette procédure explique la façon de charger un des objets et des dossiers dans un compartiment Amazon S3 à l'aide de la console.

Lorsque vous chargez un objet, le nom de clé d'objet est le nom du fichier et les préfixes facultatifs. Dans la console Amazon S3, vous pouvez créer des dossiers pour organiser vos objets. Dans Amazon S3, les dossiers sont représentés sous la forme de préfixes qui apparaissent dans le nom de la clé d'objet. Si vous téléchargez un objet individuel dans un dossier de la console Amazon S3, le nom du dossier est inclus dans le nom de la clé de l'objet.

Par exemple, si vous chargez un objet nommé `sample1.jpg` dans un dossier nommé `backup`, le nom de la clé sera `backup/sample1.jpg`. Cependant, l'objet s'affiche dans la console en tant que `sample1.jpg` dans le dossier `backup`. Pour en savoir plus sur les noms de clé, consultez Utilisation des métadonnées d'objet.

> **Note**
>
> Si vous renommez un objet ou modifiez l'une de ses propriétés dans la console Amazon S3, par exemple Classe de stockage, Chiffrement ou Métadonnées, un nouvel objet est créé pour remplacer l'ancien. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Le rôle qui modifie la propriété devient également le propriétaire du nouvel objet ou (version de l'objet).

Lorsque vous chargez un dossier, Amazon S3 charge tous les fichiers et sous-dossiers du dossier spécifié dans le compartiment. Ensuite, il attribue un nom de clé d'objet qui combine le nom du fichier chargé et le nom du dossier. Par exemple, si vous chargez un dossier nommé `/images` qui contient deux fichiers, `sample1.jpg` et `sample2.jpg`, Amazon S3 charge les fichiers, puis attribue les noms de clé correspondants, `images/sample1.jpg` et `images/sample2.jpg`. Les noms de clé incluent le nom de dossier comme préfixe. La console Amazon S3 affiche uniquement la partie du nom de clé qui suit le dernier signe `/`. Par exemple, dans un dossier `images`, les objets `images/sample1.jpg` et `images/sample2.jpg` s'affichent sous les formes `sample1.jpg` et `sample2.jpg`.

Pour charger des dossiers et des fichiers dans un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment dans lequel vous souhaitez charger vos dossiers ou fichiers.
4. Choisissez Upload.
5. Dans la fenêtre Load (Charger), procédez de l'une des manières suivantes :
   - Faites glisser et déposez les fichiers et les dossiers dans la fenêtre Upload (Charge).
   - Choisissez Ajouter un fichier ou Ajouter un dossier, puis choisissez les fichiers ou les dossiers à charger et Ouvrir.
6. Pour activer la gestion des versions, sous Destination, choisissez Enable Bucket Versioning (Activer la gestion des versions de compartiment).
7. Pour charger les fichiers et les dossiers répertoriés sans configurer des options de chargement supplémentaires, choisissez Load (Charger).

Amazon S3 charge les objets et les dossiers. Lorsque le chargement est terminé, un message de réussite s'affiche sur la page Charger : statut.

Pour configurer des propriétés d'objet supplémentaires

1. Pour modifier les autorisations de la liste de contrôle d'accès, choisissez Permissions (Autorisations).
2. Sous Access control list (ACL) (Liste de contrôle d'accès (ACL)), modifiez les autorisations.

   Pour plus d'informations sur les autorisations d'accès aux objets, consultez Utilisation de la console S3 pour définir des autorisations ACL pour un objet. Vous pouvez octroyer l'accès en lecture à vos objets au public (tout le monde) pour tous les fichiers que vous chargez. Cependant, nous recommandons de ne pas modifier le paramètre par défaut de l'accès public en lecture. L'octroi de l'accès en lecture public est applicable à un petit sous-ensemble de cas d'utilisation, comme lorsque des compartiments sont utilisés pour des sites web. Vous pouvez toujours modifier les autorisations de l'objet après l'avoir chargé.
3. Pour configurer d'autres propriétés supplémentaires, sélectionnez Properties (Propriétés).
4. Dans Classe de stockage, choisissez la classe de stockage des fichiers à charger.

   Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).
5. Pour mettre à jour les paramètres de chiffrement de vos objets, sous Server-side encryption settings (Paramètres de chiffrement côté serveur), procédez comme suit.

   a. Choisissez Specify an encryption key (Spécifier une clé de chiffrement).

   b. Sous Paramètres de chiffrement, choisissez Utiliser les paramètres du compartiment pour le chiffrement par défaut ou Ignorer les paramètres du compartiment pour le chiffrement par défaut.

   c. Si vous avez choisi Ignorer les paramètres du compartiment pour le chiffrement par défaut, vous devez configurer les paramètres de chiffrement suivants.

      - Pour chiffrer les fichiers chargés à l'aide des clés gérées par Amazon S3, choisissez Clé gérée par Amazon S3 (SSE-S3).

        Pour plus d’informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

- Pour chiffrer les fichiers téléchargés à l'aide des clés stockées dans AWS Key Management Service (AWS KMS), choisissez AWS Key Management Service key (SSE-KMS). Choisissez ensuite l'une des options suivantes pour CléAWS KMS :
  - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis sélectionnez votre Clé KMS dans la liste des clés disponibles.

    La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .
  - Pour saisir l'ARN de la clé KMS, choisissez Enter AWS KMS key ARN, puis entrez l'ARN de votre clé KMS dans le champ qui apparaît.
  - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

    Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

    > **Important**
    >
    > Vous ne pouvez utiliser que les clés KMS disponibles dans le même compartiment Région AWS que le bucket. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé KMS, puis saisir l'ARN de la clé KMS.
    > Amazon S3 prend en charge seulement les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

6. Pour utiliser des totaux de contrôle supplémentaires, sélectionnez On (Activé). Ensuite, pour le champ Checksum function (Fonction total de contrôle), sélectionnez la fonction que vous souhaitez utiliser. Amazon S3 calcule et stocke la valeur du total de contrôle après avoir reçu l'objet entier. Vous pouvez utiliser la case Precalculated value (Valeur précalculée) pour fournir

une valeur précalculée. Si vous le faites, Amazon S3 compare la valeur que vous avez fournie à la valeur qu'il calcule. Si les deux valeurs ne correspondent pas, Amazon S3 génère une erreur.

Les totaux de contrôle supplémentaires vous permettent de spécifier l'algorithme de total de contrôle que vous souhaitez utiliser pour vérifier vos données. Pour plus d'informations sur les totaux de contrôle supplémentaires, consultez Vérification de l'intégrité des objets.

7. Pour ajouter des balises à tous les objets que vous chargez, choisissez Add tag (Ajouter une balise). Saisissez un nom de balise dans le champ Clé. Saisissez une valeur pour la balise.

   Le balisage des objets vous permet de classer le stockage par catégorie. Chaque balise est une paire clés-valeurs. Les valeurs de clés et de balises sont sensibles à la casse. Vous pouvez avoir jusqu'à 10 balises par objet. Une clé de balise peut comporter jusqu'à 128 caractères Unicode et les valeurs de balise peuvent comporter jusqu'à 255 caractères Unicode. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.

8. Pour ajouter des métadonnées, choisissez Add metadata (Ajouter des métadonnées).

   a. Sous Type, choisissez System defined (Défini par le système) ou User defined (Défini par l'utilisateur).

      Pour les métadonnées définies par le système, vous pouvez sélectionner des en-têtes HTTP courants, tels que Content-Type et Content-Disposition. Pour obtenir la liste des métadonnées définies par le système et savoir si vous pouvez ajouter la valeur, veuillez consulter Métadonnées d'objet définies par le système. Toute métadonnée commençant par le préfixe `x-amz-meta-` est traitée comme une métadonnée définie par l'utilisateur. Les métadonnées définies par l'utilisateur sont stockées avec l'objet et renvoyées une fois que vous avez téléchargé l'objet. Les clés et leurs valeurs doivent respecter les normes US-ASCII. Les métadonnées définies par l'utilisateur peuvent atteindre 2 Ko. Pour plus d'informations sur les métadonnées définies par le système et par l'utilisateur, consultez Utilisation des métadonnées d'objet.

   b. Pour Key (Clé), choisissez une clé.

   c. Saisissez une valeur pour la clé.

9. Pour charger vos objets, choisissez Load (Charger).

   Amazon S3 charge votre objet. Lorsque le chargement est terminé, un message de succès s'affiche sur la page Load: status (Charger : statut).

10. Choisissez Exit (Quitter).

Utilisation des AWS SDK

Vous pouvez utiliser les AWS kits SDK pour charger des objets dans Amazon S3. Les kits SDK fournissent des bibliothèques d'enveloppe pour simplifier le chargement des données. Pour plus d'informations, consultez la liste des kits SDK pris en charge.

Voici des exemples avec quelques kits SDK sélectionnés :

.NET

L'exemple de code C# crée deux objets avec deux demandes `PutObjectRequest` :

- La première demande `PutObjectRequest`enregistre une chaîne de texte comme exemple de données d'objet. Elle spécifie aussi le nom du compartiment et celui de la clé d'objet.
- La seconde demande `PutObjectRequest` charge un fichier en spécifiant son nom. La demande spécifie aussi l'en-tête `ContentType` et les métadonnées d'objet facultatives (un titre).

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
        private const string keyName1 = "*** key name for first object created ***";
        private const string keyName2 = "*** key name for second object created
 ***";
        private const string filePath = @"*** file path ***";
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.EUWest1;
```

```
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                // 1. Put object-specify only key name for the new object.
                var putRequest1 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName1,
                    ContentBody = "sample text"
                };

                PutObjectResponse response1 = await
client.PutObjectAsync(putRequest1);

                // 2. Put the object-set ContentType and add metadata.
                var putRequest2 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName2,
                    FilePath = filePath,
                    ContentType = "text/plain"
                };

                putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
                PutObjectResponse response2 = await
client.PutObjectAsync(putRequest2);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
object"
                        , e.Message);
            }
```

```
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
 object"
                    , e.Message);
            }
        }
    }
}
```

Java

L'exemple suivant crée deux objets. Le premier objet possède une chaîne de texte comme données et le second objet est un fichier. L'exemple crée le premier objet en spécifiant le nom du compartiment, la clé d'objet et les données de texte directement dans un appel de `AmazonS3Client.putObject()`. L'exemple crée le second objet en utilisant un objet `PutObjectRequest` qui spécifie le nom du compartiment, la clé d'objet et le chemin de fichier. L'objet `PutObjectRequest` spécifie aussi l'en-tête `ContentType` et les métadonnées de titre.

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;
import java.io.IOException;

public class UploadObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
```

```
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Upload a text string as a new object.
            s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String
 Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(bucketName,
 fileObjKeyName, new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JavaScript

L'exemple suivant permet de charger un fichier existant dans un compartiment Amazon S3, dans une région spécifique.

```
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});
```

```
export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

PHP

Cet exemple vous explique comment utiliser les classes du AWS SDK for PHP pour télécharger un objet d'une taille maximale de 5 Go. Pour les fichiers plus volumineux, vous devez utiliser l'opération d'API de chargement partitionné. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Example — Création d'un objet dans un compartiment Amazon S3 en chargeant des données

L'exemple de code PHP suivant crée un objet dans un compartiment spécifié en chargeant les données grâce à la méthode `putObject()`.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);
```

```
try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'    => $keyname,
        'Body'   => 'Hello, world!',
        'ACL'    => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

La AWS SDK for Ruby version 3 propose deux méthodes pour télécharger un objet sur Amazon S3. La première utilise un chargeur de fichiers géré, qui simplifie le chargement de fichiers de toute taille à partir du disque. Pour utiliser la méthode de chargeur de fichier géré :

1. Créez une instance de la classe `Aws::S3::Resource`.
2. Référencez l'objet cible grâce au nom du compartiment et à la clé. Les objets résident dans un compartiment et disposent de clés uniques qui identifient chacun d'eux.
3. Appelez`#upload_file` sur l'objet.

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
```

```
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

La deuxième méthode utilisée par la AWS SDK for Ruby version 3 pour télécharger un objet utilise la `#put` méthode de`Aws::S3::Object`. Cette méthode est utile si l'objet est une chaîne de caractères ou un objet I/O qui n'est pas un fichier sur disque. Pour utiliser la méthode :

1. Créez une instance de la classe `Aws::S3::Resource`.
2. Référencez l'objet cible grâce au nom du compartiment et à la clé.
3. Appelez `#put` en passant la chaîne ou l'objet I/O.

Example

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
```

```
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Utilisation de l'API REST

Vous pouvez envoyer des demandes REST pour charger un objet. Vous pouvez envoyer une demande PUT pour charger des données en une seule opération. Pour plus d'informations, consultez Objet PUT.

À l'aide du AWS CLI

Vous pouvez envoyer une demande PUT pour télécharger un objet d'une taille maximale de 5 Go en une seule opération. Pour plus d'informations, veuillez consulter l'exemple `PutObject` dans la Référence de la commande AWS CLI .

Chargement et copie d'objets à l'aide d'un chargement partitionné

Le chargement partitionné vous permet de charger un seul objet en tant qu'ensemble de parties. Chaque partie est une portion contiguë des données de l'objet. Vous pouvez charger ces parties d'objet indépendamment et dans n'importe quel ordre. Si le transfert d'une partie échoue, vous pouvez la retransférer sans affecter les autres. Une fois toutes les parties de l'objet chargées, Amazon S3 les assemble et crée l'objet. En général, lorsque l'objet atteint la taille de 100 Mo, vous devez préférer les chargements partitionnés au chargement d'objet en une seule opération.

L'utilisation du chargement partitionné offre les avantages suivants :

- Improved throughput (Meilleur débit) — vous pouvez charger des parties en parallèle pour améliorer le débit.
- Quick recovery from any network issues (Récupération rapide après des problèmes réseau) — la taille réduite des parties minimise l'impact du redémarrage d'un chargement qui a échoué en raison d'une erreur de réseau.
- Pause and resume object uploads (Interruption et reprise des chargements d'objet) — vous pouvez charger des parties d'objet au fil du temps. Après le lancement d'un chargement partitionné, il n'y a aucune date d'expiration ; vous devez explicitement le terminer ou l'arrêter.
- Begin an upload before you know the final object size (Lancement d'un chargement avant de connaître la taille finale de l'objet) — vous pouvez charger un objet à mesure que vous le créez.

Nous vous recommandons d'utiliser le chargement partitionné comme suit :

- Si vous chargez des objets volumineux sur un réseau à large bande passante stable, utilisez le chargement partitionné pour optimiser l'utilisation de la bande passante disponible en chargeant des parties d'objet en parallèle pour bénéficier de performances multithreads.
- Si vous effectuez un chargement sur un réseau irrégulier, utilisez le chargement partitionné pour augmenter la résilience aux erreurs réseau en évitant les redémarrages du chargement. Lorsque vous utilisez le chargement partitionné, vous avez besoin de relancer le chargement

uniquement pour les parties d'objet dont le chargement a été interrompu. Vous n'avez pas besoin de redémarrer le chargement de vos objets depuis le début.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire. Pour plus d'informations sur l'utilisation du chargement partitionné avec S3 Express One Zone et les compartiments de répertoires, consultez Utilisation de téléchargements partitionnés avec des compartiments de répertoires.

Processus de chargement partitionné

Le chargement partitionné est un processus en trois étapes : vous lancez le chargement, vous chargez les parties de l'objet, et une fois toutes les parties chargées, vous terminez le chargement partitionné. Dès réception de la demande de fin de chargement partitionné, Amazon S3 crée l'objet à partir des parties chargées pour que vous puissiez ensuite y accéder comme vous le feriez avec n'importe quel autre objet du compartiment.

Vous pouvez lister tous vos chargements partitionnés en cours ou obtenir une liste des parties que vous avez chargées pour un chargement partitionné spécifique. Chacune de ces opérations est expliquée dans cette section.

Lancement du chargement partitionné

Lorsque vous envoyez une demande pour lancer un chargement partitionné, Amazon S3 renvoie une réponse avec un ID de chargement, qui est un identifiant unique pour le chargement partitionné. Vous devez inclure cet ID de chargement dès que vous chargez les parties, listez les parties, terminez un chargement ou arrêtez un chargement. Si vous souhaitez fournir des métadonnées qui décrivent l'objet en cours de chargement, vous devez le faire dans la demande de lancement du chargement partitionné.

Chargement de parties

Lorsque vous chargez une partie, outre l'ID de chargement, vous devez spécifier un numéro de partie. Vous pouvez choisir n'importe quel numéro de partie compris entre 1 et 10 000. Un numéro de partie identifie de manière unique une partie et sa place dans l'objet que vous chargez. Le numéro

de partie que vous choisissez ne doit pas obligatoirement constituer une séquence consécutive (par exemple, cela peut être 1, 5 et 14). Si vous chargez une nouvelle partie avec le même numéro qu'une partie précédemment chargée, cette dernière est remplacée.

Lorsque vous chargez une partie, Amazon S3 renvoie une étiquette d'entité (ETag) pour la partie sous forme d'en-tête dans la réponse. Pour chaque chargement de partie, vous devez enregistrer le numéro de partie et la valeur ETag. Vous devez inclure ces valeurs dans la demande ultérieure pour terminer le chargement partitionné. Chaque partie aura son propre ETag au moment du téléchargement. Cependant, une fois le téléchargement en plusieurs parties terminé et toutes les parties consolidées, toutes les parties seront regroupées sous un seul ETag sous forme de somme de contrôle des sommes de contrôle.

> **Note**
>
> Après avoir lancé un chargement partitionné et chargé une ou plusieurs parties, vous devez terminer ou arrêter le chargement partitionné afin que le stockage des parties chargées cesse de vous être facturé. C'est seulement après la fin ou l'arrêt d'un chargement partitionné qu'Amazon S3 libère le stockage des parties et cesse de vous le facturer.
> Après avoir arrêté un chargement partitionné, vous ne pouvez pas charger de partie avec le même ID de chargement. Si des chargements de partie étaient en cours, ils peuvent encore aboutir ou échouer, même après un arrêt du chargement. Pour être sûr de libérer tout le stockage consommé par toutes les parties, vous devez arrêter un chargement partitionné seulement après que tous les chargements partiels aient été complétés.

Fin du chargement partitionné

Lorsque vous terminez un chargement partitionné, Amazon S3 crée un objet en concaténant les parties par ordre croissant en fonction des numéros de partie. Si des métadonnées d'objet sont fournies dans la demande de lancement du chargement partitionné, Amazon S3 les associe à l'objet. À l'issue d'une demande de chargement complet, les parties n'existent plus.

La demande de chargement partitionné complet doit inclure l'ID de chargement et une liste des numéros de partie et des valeurs ETag correspondantes. La réponse d'Amazon S3 inclut une valeur ETag qui identifie de façon unique les données d'objet combinées. Cet ETag n'est pas nécessairement un hachage MD5 des données d'objet.

Exemples d'appels de chargement partitionné

Pour cet exemple, supposons que vous générez un chargement partitionné pour un fichier de 100 Go. Dans ce cas, vous recevez les appels d'API suivants pour l'ensemble du processus. Il y a un total de 1002 appels d'API.

- Appel `CreateMultipartUpload` pour commencer le processus.
- 1000 appels `UploadPart` individuels, chacun chargeant une partie de 100 Mo, pour une taille totale de 100 Go.
- Appel `CompleteMultipartUpload` pour terminer le processus.

Listes de chargement partitionné

Vous pouvez lister les parties d'un chargement partitionné spécifique ou de tous les chargements partitionnés en cours. L'opération de liste des parties renvoie des informations sur les parties que vous avez chargées pour un chargement partitionné spécifique. Pour chaque demande de liste des parties, Amazon S3 renvoie des informations sur les parties pour le chargement partitionné spécifié, pour 1 000 parties maximum. S'il y a plus de 1 000 parties dans le chargement partitionné, vous devez envoyer une série de demandes de liste des parties pour récupérer toutes les parties. Notez que la liste des parties retournée n'inclut pas les parties qui n'ont pas fini d'être chargées. En utilisant l'opération d'affichage des chargements partitionnés, vous pouvez obtenir la liste des chargements partitionnés qui sont en cours.

Un chargement partitionné en cours est un chargement que vous avez lancé, mais que vous n'avez pas encore terminé ou arrêté. Chaque demande renvoie 1,000 chargements partitionnés maximum. S'il y a plus de 1 000 chargements partitionnés en cours, vous devez envoyer des demandes supplémentaires pour récupérer les chargements partitionnés restants. Utilisez la liste renvoyée uniquement pour la vérification. N'utilisez pas le résultat de la liste lorsque vous envoyez une requête de chargement partitionné complet. Au lieu de cela, conservez votre propre liste des numéros de parties que vous avez spécifiés lors du chargement des parties ainsi que les valeurs ETag correspondantes renvoyées par Amazon S3.

Totaux de contrôle avec les opérations de chargement partitionné

Lorsque vous chargez un objet sur Simple Storage Service (Amazon S3), vous pouvez spécifier un algorithme de total de contrôle à utiliser par Amazon S3. Simple Storage Service (Amazon S3) utilise MD5 par défaut pour vérifier l'intégrité des données ; toutefois, vous pouvez spécifier un autre algorithme de total de contrôle à utiliser. Si vous utilisez MD5, Amazon S3 calcule le total de contrôle de l'ensemble de l'objet en plusieurs parties une fois le chargement terminé. Ce total de contrôle n'est

pas un total de contrôle de l'objet entier, mais plutôt un total de contrôle des totaux de contrôle de chaque partie individuelle.

Lorsque vous demandez à Amazon S3 d'utiliser des totaux de contrôle supplémentaires, Amazon S3 calcule la valeur du total de contrôle pour chaque partie et stocke les valeurs. Vous pouvez utiliser l'API ou le kit SDK pour récupérer la valeur du total de contrôle pour des parties individuelles en utilisant `GetObject` ou `HeadObject`. Pour récupérer les valeurs de contrôle des parties individuelles de téléchargements partitionnés toujours en cours, vous pouvez utiliser `ListParts`.

> **Important**
>
> Si vous utilisez un chargement partitionné avec des totaux de contrôle supplémentaires, les numéros de parties partitionnés doivent utiliser des numéros de parties consécutifs. Lorsque vous utilisez des totaux de contrôle supplémentaires, si vous essayez de compléter une requête de chargement partitionné avec des numéros de parties non consécutifs, Amazon S3 génère une erreur `500 Internal Server Error` HTTP.

Pour obtenir plus d'informations sur le fonctionnement des totaux de contrôle avec les objets en plusieurs parties, consultez Vérification de l'intégrité des objets.

Opérations simultanées de chargement partitionné

Dans un environnement de développement distribué, il est possible pour l'application de lancer plusieurs mises à jour sur le même objet en même temps. L'application doit lancer plusieurs chargements partitionnés grâce à la même clé d'objet. Pour chacun de ces chargements, l'application peut ensuite charger des parties et envoyer une demande de chargement complet à Amazon S3 pour créer l'objet. Lorsque les compartiments sont activés pour la gestion des versions S3, un chargement partitionné terminé crée toujours une nouvelle version. Pour les compartiments qui ne sont pas activés pour le contrôle de version, il est possible que d'autres demandes reçues entre le début et la fin d'un chargement partitionné priment.

> **Note**
>
> Il est possible que d'autres demandes reçues entre le début et la fin d'un chargement partitionné priment. Par exemple, si une autre opération supprime une clé entre le début et la fin d'un chargement partitionné avec cette même clé, la réponse du chargement partitionné complet peut indiquer une création d'objet réussie sans que n'ayez jamais vu l'objet.

Chargement partitionné et tarification

Lorsque vous lancez un chargement partitionné, Amazon S3 conserve toutes les parties jusqu'à ce que vous terminiez ou arrêtiez le chargement. Tout au long de sa durée de vie, le stockage, la bande passante et les demandes pour ce chargement partitionné ainsi que ses parties associées vous sont facturés.

Ces parties sont facturées en fonction de la classe de stockage spécifiée lors du chargement des parties. Les parties qui ont été chargées dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive constituent une exception à cette règle. Les parties multipartites en cours d'exécution d'une commande PUT vers la classe de stockage S3 Glacier Flexible Retrieval sont facturées en tant que stockage de transit S3 Glacier Flexible Retrieval aux tarifs de stockage S3 Standard jusqu'à ce que le chargement soit terminé. De plus, CreateMultipartUpload les deux UploadPart sont facturés aux tarifs standard S3. Seule la CompleteMultipartUpload demande est facturée au tarif S3 Glacier Flexible Retrieval. De même, les parties partitionnées en cours pour un PUT vers la classe de stockage S3 Glacier Deep Archive sont facturées en tant que stockage intermédiaire flexible S3 Glacier aux taux de stockage standard S3 jusqu'à la fin du téléchargement, seule la CompleteMultipartUpload demande étant facturée aux tarifs S3 Glacier Deep Archive.

Si vous arrêtez le chargement partitionné, Amazon S3 supprime les artefacts et les parties que vous avez chargées, et ils ne vous sont plus facturés. Aucuns frais de suppression anticipée ne sont facturés pour la suppression de chargements partitionnés incomplets, quelle que soit la classe de stockage spécifiée. Pour plus d'informations sur la tarification, consultez Tarification Amazon S3.

> **Note**
>
> Pour réduire vos coûts de stockage, nous vous recommandons de configurer une règle du cycle de vie pour supprimer les chargements partitionnés incomplets après un certain nombre de jours à l'aide de l'action `AbortIncompleteMultipartUpload`. Pour plus d'informations sur la création d'une règle de cycle de vie pour supprimer les chargements partitionnés incomplets, consultez Configuration d'une politique de cycle de vie de compartiment pour abandonner les chargements multiparties incomplets.

Prise en charge de l'API pour le chargement partitionné

Ces bibliothèques fournissent une abstraction de haut niveau qui simplifie le chargement partitionné des objets. Toutefois, si l'application l'exige, vous pouvez utiliser directement l'API REST. Les

sections suivantes de la Référence de l'API Amazon Simple Storage Service décrivent l'API REST pour le chargement partitionné.

Pour une procédure de téléchargement en plusieurs parties utilisant les fonctions AWS Lambda, consultez la section Chargement d'objets volumineux sur Amazon S3 à l'aide du téléchargement en plusieurs parties et de l'accélération du transfert.

- Création d'un chargement partitionné
- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Interruption du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

AWS Command Line Interface support pour le téléchargement partitionné

Les rubriques suivantes AWS Command Line Interface décrivent les opérations de téléchargement partitionné.

- Lancement du chargement partitionné
- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Interruption du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

AWS Support du SDK pour le téléchargement en plusieurs parties

Vous pouvez utiliser un AWS SDK pour télécharger un objet en plusieurs parties. Pour obtenir la liste des AWS SDK pris en charge par l'action de l'API, voir :

- Création d'un chargement partitionné

- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Interruption du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

API de chargement partitionné et autorisations

Vous devez posséder les autorisations nécessaires pour utiliser les opérations de chargement partitionné. Vous pouvez utiliser les listes ACL, la stratégie de compartiment ou la stratégie d'utilisateur pour autoriser des individus à exécuter ces opérations. Le tableau suivant liste les autorisations nécessaires pour les différentes opérations de chargement partitionné si vous utilisez des listes ACL, une stratégie de compartiment ou une stratégie d'utilisateur.

Action	Autorisations requises
Créer un chargement partitionné	Vous devez être autorisé à exécuter l'action `s3:PutObject` sur un objet pour créer un chargement partitionné. Le propriétaire du compartiment peut autoriser d'autres personnes habilitées à exécuter l'action `s3:PutObject` .
Lancement du chargement partitionné	Vous devez être autorisé à exécuter l'action `s3:PutObject` sur un objet pour lancer le chargement partitionné. Le propriétaire du compartiment peut autoriser d'autres personnes habilitées à exécuter l'action `s3:PutObject` .
Initiateur	Elément du conteneur qui identifie l'initiateur du chargement partitionné. Si l'initiateur est un Compte AWS, cet élément fournit les mêmes informations que l'élément Owner. Si l'initiateur est un utilisateur IAM, cet élément fournit l'ARN utilisateur et le nom complet.
Chargement d'une partie	Vous devez être autorisé à exécuter l'action `s3:PutObject` sur un objet pour charger une partie.

Action	Autorisations requises
	Le propriétaire du compartiment doit autoriser l'initiateur à exécuter l'action `s3:PutObject` sur un objet pour que ce dernier charge une partie pour cet objet.
Chargement d'une partie (Copy)	Vous devez être autorisé à exécuter l'action `s3:PutObject` sur un objet pour charger une partie. Sachant que vous chargez une partie d'un objet existant, vous devez être autorisé à exécuter l'action `s3:GetObject` sur l'objet source. Pour que l'initiateur puisse charger une partie pour un objet, le propriétaire du compartiment doit l'autoriser à effectuer l'action `s3:PutObject` sur l'objet.
Achèvement du chargement partitionné	Vous devez être autorisé à exécuter l'action `s3:PutObject` sur un objet pour terminer le chargement partitionné. Le propriétaire du compartiment doit autoriser l'initiateur à exécuter l'action `s3:PutObject` sur un objet pour que ce dernier termine un chargement partitionné pour cet objet.
Arrêt du chargement partitionné	Vous devez être autorisé à exécuter l'action `s3:AbortMultipartUpload` pour arrêter un chargement partitionné. Par défaut, le propriétaire du compartiment et l'initiateur du chargement partitionné sont autorisés à exécuter cette action dans le cadre des politiques IAM et de compartiment. Si l'initiateur est un utilisateur IAM, celui-ci Compte AWS est également autorisé à arrêter ce téléchargement partitionné. Dans le cadre des politiques de point de terminaison de VPC, l'initiateur du chargement partitionné n'a pas automatiquement l'autorisation d'effectuer l'action `s3:AbortMultipartUpload` . Outre ces paramètres par défaut, le propriétaire du compartiment peut autoriser d'autres principaux à être habilité à exécuter l'action `s3:AbortMultipartUpload` sur un objet. Le propriétaire du compartiment peut refuser que tout principal soit habilité à exécuter l'action `s3:AbortMultipartUpload` .

Action	Autorisations requises
Liste des parties	Vous devez être autorisé à exécuter l'action `s3:ListMultipartUploadParts` sur un objet pour lister les parties lors d'un chargement partitionné. Par défaut, le propriétaire du compartiment est autorisé à lister les parties pour tout chargement partitionné dans le compartiment. L'initiateur du chargement partitionné est autorisé à lister les parties du chargement partitionné spécifique. Si l'initiateur du téléchargement partitionné est un utilisateur IAM, l'utilisateur IAM Compte AWS contrôlant cet utilisateur est également autorisé à répertorier les parties de ce téléchargement. Outre ces paramètres par défaut, le propriétaire du compartiment peut autoriser d'autres principaux à être habilité à exécuter l'action `s3:ListMultipartUploadParts` sur un objet. Le propriétaire du compartiment peut également refuser que tout principal soit habilité à exécuter l'action `s3:ListMultipartUploadParts` .
Liste des chargements partitionnés	Vous devez être autorisé à exécuter l'action `s3:ListBucketMultipartUploads` sur un compartiment pour lister les chargements partitionnés en cours dans ce compartiment. Outre ces paramètres par défaut, le propriétaire du compartiment peut autoriser d'autres personnes habilitées à exécuter l'action `s3:ListBucketMultipartUploads` sur le compartiment.

Action	Autorisations requises
AWS KMS Chiffrer et déchiffrer les autorisations associées	Pour effectuer un téléchargement partitionné avec chiffrement à l'aide d'une clé KMS AWS Key Management Service (AWS KMS), le demandeur doit être autorisé à effectuer les `kms:GenerateDataKey` actions `kms:Decrypt` et sur la clé. Ces autorisations sont requises, car Simple Storage Service (Amazon S3) doit déchiffrer et lire les données des parties de fichier chiffrées avant de terminer le chargement partitionné. Pour plus d'informations, consultez Chargement d'un fichier volumineux vers Amazon S3 avec chiffrement à l'aide d'une AWS KMS key que vous trouverez dans le AWS Centre de connaissances. Si votre utilisateur ou rôle IAM est Compte AWS identique à celui de la clé KMS, vous devez disposer de ces autorisations sur la politique de clé. Si votre utilisate ur ou rôle IAM appartient à un autre compte que la clé KMS, vous devez disposer des autorisations sur la politique de clé et votre utilisateur ou rôle IAM.

Pour en savoir plus sur la relation entre les autorisations de liste ACL et les autorisations des politiques d'accès, consultez Mappage des autorisations de liste ACL et de stratégie d'accès. Pour obtenir plus d'informations sur les utilisateurs IAM, les rôles et les bonnes pratiques, consultez la section IAM Identities (users, user groups, and roles) [Identités IAM (utilisateurs, groupes d'utilisateurs et rôles)] dans le Guide de l'utilisateur IAM.

Rubriques

- Configuration d'une configuration de cycle de vie de compartiment pour supprimer les chargements partitionnés incomplets
- Chargement d'un objet à l'aide du chargement partitionné
- Téléchargement d'un répertoire à l'aide de la classe .NET TransferUtility de haut niveau
- Liste des chargements partitionnés
- Suivi d'un chargement partitionné
- Interruption d'un chargement partitionné
- Copie d'un objet à l'aide du chargement partitionné
- Limites de la fonction de chargement partitionné Amazon S3

Configuration d'une configuration de cycle de vie de compartiment pour supprimer les chargements partitionnés incomplets

Dans le cadre des bonnes pratiques, nous vous recommandons de configurer une règle du cycle de vie à l'aide de l'action `AbortIncompleteMultipartUpload` pour réduire les coûts de stockage. Pour plus d'informations sur l'abandon d'un chargement multipartie, consultez Interruption d'un chargement partitionné.

Simple Storage Service (Amazon S3) prend en charge une règle de cycle de vie de compartiment que vous pouvez utiliser pour demander à Simple Storage Service (Amazon S3) d'arrêter les chargements partitionnés s'ils ne se terminent pas avant la valeur spécifiée pour le nombre de jours après leur lancement. Lorsqu'un chargement partitionné n'est pas terminé dans le délai imparti, il peut être annulé. Amazon S3 arrête alors le chargement partitionné et supprime les pièces associées au chargement partitionné. Cette règle s'applique à la fois aux téléchargements partitionnés existants et à ceux que vous créerez ultérieurement.

Voici un exemple de configuration du cycle de vie qui spécifie une règle avec l'action `AbortIncompleteMultipartUpload`.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Dans l'exemple, la règle ne spécifie pas de valeur pour l'élément `Prefix` (le préfixe du nom de la clé de l'objet). Par conséquent, la règle s'applique à tous les objets du compartiment pour lesquels vous avez lancé des téléchargements partitionnés. Tous les téléchargements partitionnés qui ont été lancés et qui ne se sont pas terminés dans les sept jours peuvent faire l'objet d'une interruption. L'action Annuler n'a aucun effet sur les téléchargements partitionnés terminés.

Pour en savoir plus sur la configuration du cycle de vie du compartiment, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Si le chargement partitionné est terminé dans le temps imparti spécifié dans la règle, l'action du cycle de vie `AbortIncompleteMultipartUpload` ne s'applique pas (cela signifie qu'Amazon S3 n'applique aucune action). En outre, cette action ne s'applique pas aux objets. Aucun objet n'est supprimé par cette action de cycle de vie. De plus, vous n'aurez pas à payer de frais de suppression anticipée pour le cycle de vie S3 lorsque vous supprimez des parties de chargements partitionnés incomplets.

Utilisation de la console S3

Pour gérer automatiquement les chargements partitionnés incomplets, vous pouvez utiliser la console S3 pour créer une règle de cycle de vie afin de faire expirer les octets de chargement partitionnés incomplets de votre compartiment après un nombre de jours spécifié. La procédure suivante vous montre comment ajouter une règle de cycle de vie pour supprimer les chargements partitionnés incomplets après sept jours. Pour plus d'informations sur l'ajout de règles de cycle de vie, consultez Configuration du cycle de vie d'un bucket.

Pour ajouter une règle de cycle de vie afin d'interrompre les chargements partitionnés incomplets datant de plus de sept jours.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez créer une stratégie de cycle de vie.
3. Choisissez l'onglet Management (Gestion), puis choisissez Create lifecycle rule (Créer une règle de cycle de vie).
4. Dans Lifecycle rule name (Nom de la règle du cycle de vie), saisissez un nom pour votre règle.

   Ce nom doit être unique dans le compartiment.
5. Choisissez l'étendue de la règle de cycle de vie :

   - Pour créer une règle de cycle de vie pour tous les objets ayant un préfixe spécifique, sélectionnez Limit the scope of this rule using one or more filters (Limiter l'étendue de cette règle à l'aide d'un ou plusieurs filtres), puis saisissez le préfixe dans le champ Prefix (Préfixe).

- Pour créer une règle de cycle de vie pour tous les objets du compartiment, sélectionnez l'option This rule applies to all objects in the bucket (Cette règle s'applique à tous les objets du compartiment), puis sélectionnez I acknowledge that this rule applies to all objects in the bucket (Je reconnais que cette règle s'applique à tous les objets du compartiment).

6. Sous Lifecycle rule actions (Actions des règles de cycle de vie), sélectionnez Delete expired object delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression d'objets expirés ou les chargements partitionnés incomplets).
7. Sous Delete expired object delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression des objets expirés ou les chargements partitionnés incomplets), sélectionnez Delete incomplete multipart uploads (Supprimer les chargements partitionnés incomplets).
8. Dans le champ Number of days (Nombre de jours), saisissez le nombre de jours au bout duquel vous souhaitez supprimer les chargements partitionnés incomplets (dans cet exemple, sept jours).
9. Choisissez Créer une règle.

À l'aide du AWS CLI

La commande suivante `put-bucket-lifecycle-configuration`AWS Command Line Interface (AWS CLI) ajoute la configuration du cycle de vie pour le compartiment spécifié. Pour utiliser cette commande, remplacez *`user input placeholders`* par vos informations.

```
aws s3api put-bucket-lifecycle-configuration  \
        --bucket example-s3-bucket1  \
        --lifecycle-configuration filename-containing-lifecycle-configuration
```

L'exemple suivant montre comment ajouter une règle de cycle de vie pour interrompre les chargements partitionnés incomplets en utilisant la AWS CLI. Il comprend un exemple de configuration du cycle de vie JSON pour interrompre les chargements partitionnés incomplets datant de plus de sept jours.

Pour utiliser les commandes CLI de cet exemple, remplacez *`user input placeholders`* par vos informations.

Pour ajouter une règle de cycle de vie afin d'interrompre les chargements partitionnés incomplets

1. Configurez le AWS CLI. Pour obtenir des instructions, veuillez consulter Développement avec Amazon S3 à l'aide de la AWS CLI.
2. Enregistrez l'exemple de configuration du cycle de vie suivant dans un fichier (par exemple, *lifecycle.json*). Cet exemple de configuration spécifie un préfixe vide, et s'applique donc à tous les objets du compartiment. Pour restreindre la configuration à un sous-ensemble d'objets, vous pouvez spécifier un préfixe.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": ""
            },
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
}
```

3. Exécutez la commande CLI suivante pour configurer cette configuration du cycle de vie sur le compartiment.

```
aws s3api put-bucket-lifecycle-configuration   \
--bucket example-s3-bucket1  \
--lifecycle-configuration file://lifecycle.json
```

4. Pour vérifier que la configuration du cycle de vie a été définie sur votre compartiment, récupérez la configuration du cycle de vie en utilisant la commande `get-bucket-lifecycle` suivante.

```
aws s3api get-bucket-lifecycle  \
--bucket example-s3-bucket1
```

5. Pour supprimer la configuration du cycle de vie, utilisez la commande `delete-bucket-lifecycle` suivante.

```
aws s3api delete-bucket-lifecycle \
```

```
--bucket example-s3-bucket1
```

Chargement d'un objet à l'aide du chargement partitionné

Vous pouvez utiliser le chargement partitionné pour charger par programme un seul objet sur Amazon S3.

Pour plus d'informations, consultez les sections suivantes.

Utilisation des AWS SDK (API de haut niveau)

Certains AWS SDK proposent une API de haut niveau qui simplifie le téléchargement en plusieurs parties en combinant les différentes opérations d'API requises pour effectuer un téléchargement en plusieurs parties en une seule opération. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

Si vous devez suspendre et reprendre les téléchargements partitionnés, faire varier la taille des parties pendant le téléchargement ou si vous ne connaissez pas la taille des données à l'avance, utilisez les méthodes d'API de bas niveau. Les méthodes d'API de bas niveau pour les téléchargements partitionnés offrent des fonctionnalités supplémentaires. Pour plus d'informations, voir. Utilisation des AWS SDK (API de bas niveau)

Java

Pour télécharger des fichiers volumineux, utilisez la `TransferManager` classe. Cette opération d'API de haut niveau permet de télécharger des données à partir d'un fichier ou d'un flux. Vous pouvez également définir des options avancées, telles que la taille du composant que vous souhaitez utiliser pour le chargement partitionné ou le nombre de threads simultanés que vous voulez utilisez lors du chargement des composants. Vous pouvez également définir des propriétés d'objet optionnelles, la classe de stockage ou la liste de contrôle d'accès (ACL). Vous utilisez les classes `PutObjectRequest` et `TransferManagerConfiguration` pour définir ces options avancées.

Dans la mesure du possible, `TransferManager` essaie d'utiliser des threads multiples pour charger plusieurs parties d'un chargement unique en une seule fois. Ceci peut améliorer le débit de façon significative lorsque vous gérez de gros volumes de contenu et une bande passante importante.

En plus de la fonctionnalité de chargement de fichier, la classe `TransferManager` vous permet d'arrêter un chargement partitionné en cours. Un chargement est considéré en cours dès que

vous le lancez et jusqu'à ce qu'il soit terminé ou arrêté. `TransferManager` arrête tous les chargements partitionnés en cours sur un compartiment spécifié qui ont été lancés avant une date et une heure données.

> **Note**
>
> Lorsque vous utilisez un flux comme source de données, la classe `TransferManager` n'effectue pas de chargements simultanés.

L'exemple suivant charge un objet à l'aide de l'API Java de haut niveau de chargements partitionnés (la classe `TransferManager`). Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import java.io.File;

public class HighLevelMultipartUpload {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** Path for file to upload ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();
            TransferManager tm = TransferManagerBuilder.standard()
```

```
                    .withS3Client(s3Client)
                    .build();

            // TransferManager processes all transfers asynchronously,
            // so this call returns immediately.
            Upload upload = tm.upload(bucketName, keyName, new File(filePath));
            System.out.println("Object upload started");

            // Optionally, wait for the upload to finish before continuing.
            upload.waitForCompletion();
            System.out.println("Object upload complete");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Pour charger un fichier dans un compartiment S3, utilisez la classe `TransferUtility`. Lorsque vous chargez les données d'un fichier, vous devez fournir le nom de clé de l'objet. Sinon, l'API utilise le nom du fichier comme nom de clé. Lorsque vous chargez les données d'un flux, vous devez fournir le nom de clé de l'objet.

Utilisez la classe `TransferUtilityUploadRequest` pour définir des options avancées, comme la taille des parties, le nombre de threads lors du chargement simultané des parties, les métadonnées, la classe de stockage ou une liste ACL.

> **Note**
>
> Lorsque vous utilisez un flux comme source de données, la classe `TransferUtility` n'effectue pas de chargements simultanés.

L'exemple C# suivant charge un fichier dans un compartiment Amazon S3 en plusieurs parties. Il montre comment utiliser différentes surcharges `TransferUtility.Upload` pour charger un fichier. Chaque appel de chargement successif remplace le chargement précédent. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
 ***";
        private const string filePath = "*** provide the full path name of the file
 to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadFileAsync().Wait();
        }

        private static async Task UploadFileAsync()
        {
            try
            {
                var fileTransferUtility =
                    new TransferUtility(s3Client);

                // Option 1. Upload a file. The file name is used as the object key
 name.
                await fileTransferUtility.UploadAsync(filePath, bucketName);
```

```
                Console.WriteLine("Upload 1 completed");

                // Option 2. Specify object key name explicitly.
                await fileTransferUtility.UploadAsync(filePath, bucketName,
keyName);
                Console.WriteLine("Upload 2 completed");

                // Option 3. Upload data from a type of System.IO.Stream.
                using (var fileToUpload =
                    new FileStream(filePath, FileMode.Open, FileAccess.Read))
                {
                    await fileTransferUtility.UploadAsync(fileToUpload,
                                                bucketName, keyName);
                }
                Console.WriteLine("Upload 3 completed");

                // Option 4. Specify advanced settings.
                var fileTransferUtilityRequest = new TransferUtilityUploadRequest
                {
                    BucketName = bucketName,
                    FilePath = filePath,
                    StorageClass = S3StorageClass.StandardInfrequentAccess,
                    PartSize = 6291456, // 6 MB.
                    Key = keyName,
                    CannedACL = S3CannedACL.PublicRead
                };
                fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
                fileTransferUtilityRequest.Metadata.Add("param2", "Value2");

                await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
                Console.WriteLine("Upload 4 completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }

        }
```

```
    }
}
```

JavaScript

Example

Chargez un fichier volumineux.

```
import {
  CreateMultipartUploadCommand,
  UploadPartCommand,
  CompleteMultipartUploadCommand,
  AbortMultipartUploadCommand,
  S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

export const main = async () => {
  const s3Client = new S3Client({});
  const bucketName = "test-bucket";
  const key = "multipart.txt";
  const str = createString();
  const buffer = Buffer.from(str, "utf8");

  let uploadId;

  try {
    const multipartUpload = await s3Client.send(
      new CreateMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
      }),
    );

    uploadId = multipartUpload.UploadId;

    const uploadPromises = [];
    // Multipart uploads require a minimum size of 5 MB per part.
```

```
  const partSize = Math.ceil(buffer.length / 5);

  // Upload each part.
  for (let i = 0; i < 5; i++) {
    const start = i * partSize;
    const end = start + partSize;
    uploadPromises.push(
      s3Client
        .send(
          new UploadPartCommand({
            Bucket: bucketName,
            Key: key,
            UploadId: uploadId,
            Body: buffer.subarray(start, end),
            PartNumber: i + 1,
          }),
        )
        .then((d) => {
          console.log("Part", i + 1, "uploaded");
          return d;
        }),
    );
  }

  const uploadResults = await Promise.all(uploadPromises);

  return await s3Client.send(
    new CompleteMultipartUploadCommand({
      Bucket: bucketName,
      Key: key,
      UploadId: uploadId,
      MultipartUpload: {
        Parts: uploadResults.map(({ ETag }, i) => ({
          ETag,
          PartNumber: i + 1,
        })),
      },
    }),
  );

  // Verify the output by downloading the file from the Amazon Simple Storage
Service (Amazon S3) console.
  // Because the output is a 25 MB string, text editors might struggle to open the
file.
```

```
  } catch (err) {
    console.error(err);

    if (uploadId) {
      const abortCommand = new AbortMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
        UploadId: uploadId,
      });

      await s3Client.send(abortCommand);
    }
  }
};
```

Example

Téléchargez un fichier volumineux.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: parseInt(start),
    end: parseInt(end),
```

```
    length: parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Go

Example

Chargez un objet volumineux à l'aide d'un gestionnaire de chargement qui divise les données en plusieurs parties et les charge simultanément.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3) actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}
```

```
// UploadLargeObject uses an upload manager to upload data to an object in a bucket.
// The upload manager breaks large data into parts and uploads the parts
 concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
 largeObject []byte) error {
 largeBuffer := bytes.NewReader(largeObject)
 var partMiBs int64 = 10
 uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
  u.PartSize = partMiBs * 1024 * 1024
 })
 _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
  Body:   largeBuffer,
 })
 if err != nil {
  log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }

 return err
}
```

Example

Téléchargez un objet volumineux en utilisant un gestionnaire de téléchargement pour obtenir les données en plusieurs parties et les télécharger simultanément.

```
// DownloadLargeObject uses a download manager to download an object from a bucket.
// The download manager gets the data in parts and writes them to a buffer until all
 of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey string)
 ([]byte, error) {
 var partMiBs int64 = 10
 downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader) {
  d.PartSize = partMiBs * 1024 * 1024
 })
 buffer := manager.NewWriteAtBuffer([]byte{})
 _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return buffer.Bytes(), err
}
```

PHP

Cette rubrique explique comment utiliser la `Aws\S3\Model\MultipartUpload\UploadBuilder` classe de haut niveau de la AWS SDK for PHP pour les téléchargements de fichiers en plusieurs parties. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple de code PHP ci-dessous permet de charger un fichier vers un compartiment Amazon S3. Il explique comment définir les paramètres de l'objet `MultipartUploader`.

```
 require 'vendor/autoload.php';

use Aws\Exception\MultipartUploadException;
```

```
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'    => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Python

L'exemple suivant charge un objet à l'aide de l'API Python de haut niveau de chargements partitionnés (la classe `TransferManager`).

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig


MB = 1024 * 1024
s3 = boto3.resource("s3")


class TransferCallback:
```

```
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()


def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
```

```
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info


def upload_with_high_threshold(local_file_path, bucket_name, object_key,
 file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
```

```
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
 Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Callback=transfer_callback
    )
```

```
    return transfer_callback.thread_info


def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.
```

```
    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

Utilisation des AWS SDK (API de bas niveau)

Le AWS SDK présente une API de bas niveau qui ressemble beaucoup à l'API REST d'Amazon S3 pour les téléchargements partitionnés (voir. Chargement et copie d'objets à l'aide d'un chargement partitionné Utilisez l'API de bas niveau lorsque vous devez suspendre et reprendre les téléchargements partitionnés, faire varier la taille des parties pendant le téléchargement ou si vous ne connaissez pas la taille des données de téléchargement à l'avance. Lorsque vous n'avez pas ces exigences, utilisez l'API de haut niveau (voirUtilisation des AWS SDK (API de haut niveau)).

Java

L'exemple suivant montre comment utiliser les classes Java de bas niveau pour charger un fichier. Il exécute les étapes suivantes :

- Initie un chargement partitionné à l'aide de la méthode `AmazonS3Client.initiateMultipartUpload()` et transmet un objet `InitiateMultipartUploadRequest`.
- Enregistre l'ID de chargement renvoyé par la méthode `AmazonS3Client.initiateMultipartUpload()`. Vous devez fournir cet ID de chargement pour chaque opération suivante de chargement partitionné.
- Charge les parties de l'objet. Pour chaque partie, appelez la méthode `AmazonS3Client.uploadPart()`. Vous fournissez les informations de chargement de partie à l'aide d'un objet `UploadPartRequest`.

- Pour chaque partie, enregistre l'ETag de la réponse de la méthode `AmazonS3Client.uploadPart()` dans une liste. Vous utilisez les valeurs ETag pour terminer le chargement partitionné.
- Appelle la méthode `AmazonS3Client.completeMultipartUpload()` pour terminer le chargement partitionné.

Example

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartUpload {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String filePath = "*** Path to file to upload ***";

        File file = new File(filePath);
        long contentLength = file.length();
        long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();
```

```
            // Create a list of ETag objects. You retrieve ETags for each object
part
            // uploaded,
            // then, after each individual part has been uploaded, pass the list of
ETags to
            // the request to complete the upload.
            List<PartETag> partETags = new ArrayList<PartETag>();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(bucketName, keyName);
            InitiateMultipartUploadResult initResponse =
s3Client.initiateMultipartUpload(initRequest);

            // Upload the file parts.
            long filePosition = 0;
            for (int i = 1; filePosition < contentLength; i++) {
                // Because the last part could be less than 5 MB, adjust the part
size as
                // needed.
                partSize = Math.min(partSize, (contentLength - filePosition));

                // Create the request to upload a part.
                UploadPartRequest uploadRequest = new UploadPartRequest()
                        .withBucketName(bucketName)
                        .withKey(keyName)
                        .withUploadId(initResponse.getUploadId())
                        .withPartNumber(i)
                        .withFileOffset(filePosition)
                        .withFile(file)
                        .withPartSize(partSize);

                // Upload the part and add the response's ETag to our list.
                UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
                partETags.add(uploadResult.getPartETag());

                filePosition += partSize;
            }

            // Complete the multipart upload.
            CompleteMultipartUploadRequest compRequest = new
CompleteMultipartUploadRequest(bucketName, keyName,
                    initResponse.getUploadId(), partETags);
```

```
            s3Client.completeMultipartUpload(compRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

L'exemple C# suivant montre comment utiliser l'API de téléchargement AWS SDK for .NET partitionné de bas niveau pour télécharger un fichier dans un compartiment S3. Pour plus d'informations sur les chargements partitionnés Amazon S3, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

> **Note**
>
> Lorsque vous utilisez l' AWS SDK for .NET API pour télécharger des objets volumineux, un délai d'attente peut se produire lors de l'écriture des données dans le flux de demandes. Vous pouvez définir un délai explicite en utilisant `UploadPartRequest`.

L'exemple de code C# suivant charge un fichier dans un compartiment S3 grâce à l'API de chargement partitionné de bas niveau. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
 ***";
        private const string filePath = "*** provide the full path name of the file
 to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Uploading an object");
            UploadObjectAsync().Wait();
        }

        private static async Task UploadObjectAsync()
        {
            // Create list to store upload part responses.
            List<UploadPartResponse> uploadResponses = new
 List<UploadPartResponse>();

            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest = new
 InitiateMultipartUploadRequest
            {
                BucketName = bucketName,
                Key = keyName
            };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Upload parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
```

```
            try
            {
                Console.WriteLine("Uploading parts");

                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                        {
                            BucketName = bucketName,
                            Key = keyName,
                            UploadId = initResponse.UploadId,
                            PartNumber = i,
                            PartSize = partSize,
                            FilePosition = filePosition,
                            FilePath = filePath
                        };

                    // Track upload progress.
                    uploadRequest.StreamTransferProgress +=
                        new
EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

                    // Upload a part and add the response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }

                // Setup to complete the upload.
                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        UploadId = initResponse.UploadId
                     };
                completeRequest.AddPartETags(uploadResponses);

                // Complete the upload.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
```

```
            catch (Exception exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown: { 0}",
 exception.Message);

                // Abort the upload.
                AbortMultipartUploadRequest abortMPURequest = new
 AbortMultipartUploadRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    UploadId = initResponse.UploadId
                };
               await s3Client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
        public static void UploadPartProgressEventCallback(object sender,
 StreamTransferProgressArgs e)
        {
            // Process event.
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

PHP

Cette rubrique explique comment utiliser la `uploadPart` méthode de bas niveau de la version 3 de AWS SDK for PHP pour télécharger un fichier en plusieurs parties. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple PHP suivant charge un fichier dans un compartiment Amazon S3 grâce au chargement partitionné de l'API PHP de bas niveau.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';
```

```
$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket'       => $bucket,
    'Key'          => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata'     => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket'     => $bucket,
            'Key'        => $keyname,
            'UploadId'   => $uploadId,
            'PartNumber' => $partNumber,
            'Body'       => fread($file, 5 * 1024 * 1024),
        ]);
        $parts['Parts'][$partNumber] = [
            'PartNumber' => $partNumber,
            'ETag' => $result['ETag'],
        ];
        $partNumber++;

        echo "Uploading part $partNumber of $filename." . PHP_EOL;
    }
    fclose($file);
} catch (S3Exception $e) {
    $result = $s3->abortMultipartUpload([
        'Bucket'   => $bucket,
        'Key'      => $keyname,
        'UploadId' => $uploadId
    ]);
```

```
    echo "Upload of $filename failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
    'MultipartUpload'    => $parts,
]);
$url = $result['Location'];

echo "Uploaded $filename to $url." . PHP_EOL;
```

À l'aide du AWS SDK for Ruby

La AWS SDK for Ruby version 3 prend en charge les téléchargements partitionnés sur Amazon S3 de deux manières. Pour la première option, vous pouvez utiliser des chargements de fichiers gérés. Pour plus d'informations, consultez Uploading Files to Amazon S3 dans le Blog des développeurs AWS . Les chargements de fichiers gérés constituent la méthode recommandée pour charger des fichiers dans un compartiment. Ces chargements offrent les avantages suivants :

- Ils gèrent les chargements partitionnés pour les objets de plus de 15 Mo.
- Ils ouvrent correctement les fichiers en mode binaire pour éviter les problèmes d'encodage.
- Ils utilisent plusieurs threads pour le chargement des parties d'objets volumineux en parallèle.

Vous pouvez également utiliser directement les opérations suivantes du client de chargement partitionné :

- create_multipart_upload – Lance un chargement partitionné et renvoie un ID de chargement.
- upload_part – Charge une partie d'un chargement partitionné.
- upload_part_copy – Charge une partie en copiant les données d'un objet existant comme source de données.
- complete_multipart_upload – Termine un chargement partitionné en assemblant des parties chargées précédemment.
- abort_multipart_upload – Arrête un chargement partitionné.

Utilisation de l'API REST

Les sections suivantes de la Référence de l'API Amazon Simple Storage Service décrivent l'API REST pour le chargement partitionné.

- Lancement du chargement partitionné
- Chargement d'une partie
- Achèvement du chargement partitionné
- Arrêt du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

À l'aide du AWS CLI

Les sections suivantes du AWS Command Line Interface (AWS CLI) décrivent les opérations de téléchargement partitionné.

- Lancement du chargement partitionné
- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Interruption du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

Vous pouvez également utiliser l'API REST pour vos propres demandes REST ou l'un des kits AWS SDK. Pour plus d'informations sur l'API REST, consultez Utilisation de l'API REST. Pour plus d'informations sur les kits SDK, consultez Chargement d'un objet à l'aide du chargement partitionné.

Téléchargement d'un répertoire à l'aide de la classe .NET TransferUtility de haut niveau

Vous pouvez utiliser la classe `TransferUtility` pour charger un répertoire entier. Par défaut, l'API charge uniquement les fichiers à la racine du répertoire spécifié. Toutefois, vous pouvez spécifier de charger les fichiers de manière récursive dans tous les sous-répertoires.

Indiquez des expressions de filtre pour sélectionner des fichiers dans le répertoire spécifié selon des critères de filtrage. Par exemple, pour ne charger que les fichiers .pdf d'un répertoire, spécifiez l'expression de filtre `"*.pdf"`.

Quand vous chargez des données depuis un répertoire, vous ne spécifiez pas les noms de clé pour les objets résultants. Amazon S3 génère les noms de clé à l'aide du chemin du fichier d'origine. Par exemple, si vous avez un répertoire appelé `c:\myfolder` avec la structure suivante :

Example

```
C:\myfolder
      \a.txt
      \b.pdf
      \media\
             An.mp3
```

Lorsque vous chargez ce répertoire, Amazon S3 utilise les noms de clé suivants :

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

L'exemple C# suivant charge un répertoire dans un compartiment Amazon S3. Il montre comment utiliser différentes surcharges `TransferUtility.UploadDirectory` pour charger le répertoire. Chaque appel de chargement successif remplace le chargement précédent. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
```

```
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string directoryPath = @"*** directory path ***";
        // The example uploads only .txt files.
        private const string wildCard = "*.txt";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadDirAsync().Wait();
        }

        private static async Task UploadDirAsync()
        {
            try
            {
                var directoryTransferUtility =
                    new TransferUtility(s3Client);

                // 1. Upload a directory.
                await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
                    existingBucketName);
                Console.WriteLine("Upload statement 1 completed");

                // 2. Upload only the .txt files from a directory
                //    and search recursively.
                await directoryTransferUtility.UploadDirectoryAsync(
                                                directoryPath,
                                                existingBucketName,
                                                wildCard,
                                                SearchOption.AllDirectories);
                Console.WriteLine("Upload statement 2 completed");

                // 3. The same as Step 2 and some optional configuration.
                //    Search recursively for .txt files to upload.
                var request = new TransferUtilityUploadDirectoryRequest
                {
                    BucketName = existingBucketName,
                    Directory = directoryPath,
                    SearchOption = SearchOption.AllDirectories,
                    SearchPattern = wildCard
                };
```

```
                await directoryTransferUtility.UploadDirectoryAsync(request);
                Console.WriteLine("Upload statement 3 completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an object",
 e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
 object", e.Message);
            }
        }
    }
}
```

Liste des chargements partitionnés

Vous pouvez utiliser les AWS SDK (API de bas niveau) pour récupérer une liste des téléchargements partitionnés en cours dans Amazon S3.

Répertorier les téléchargements partitionnés à l'aide du AWS SDK (API de bas niveau)

Java

Les tâches suivantes vous guident dans l'utilisation des classes Java de bas niveau pour répertorier tous les chargements partitionnés en cours sur un compartiment.

Processus d'élaboration de la liste des chargements partitionnés via l'API de bas niveau

1	Créez une instance de la classe `ListMultipartUploadsRequest` et précisez le nom du compartiment.
2	Exécutez la méthode `AmazonS3Client.listMultipartUploads`. Cette méthode renvoie une instance de la classe `MultipartUploadListing`, qui vous fournit des informations sur les chargements partitionnés en cours.

L'exemple de code Java suivant présente les tâches précédentes.

Example

```
ListMultipartUploadsRequest allMultpartUploadsRequest =
     new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
     s3Client.listMultipartUploads(allMultpartUploadsRequest);
```

.NET

Pour afficher tous les chargements partitionnés en cours vers un compartiment spécifique, utilisez la classe `ListMultipartUploadsRequest` de l'API de chargement partitionné de bas niveau du kit AWS SDK for .NET . La méthode `AmazonS3Client.ListMultipartUploads` retourne une instance de la classe `ListMultipartUploadsResponse` qui fournit des informations sur les chargements partitionnés en cours.

Un chargement partitionné en cours est un chargement partitionné qui a été lancé à l'aide d'une demande de lancement de chargement partitionné, mais qui n'a pas encore été terminé ou arrêté. Pour en savoir plus sur les chargements partitionnés Amazon S3, veuillez consulter Chargement et copie d'objets à l'aide d'un chargement partitionné.

L'exemple C# suivant montre comment utiliser le pour AWS SDK for .NET répertorier tous les téléchargements partitionnés en cours sur un bucket. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
  BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
 AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

Cette rubrique explique comment utiliser les classes d'API de bas niveau de la version 3 de AWS SDK for PHP pour répertorier tous les téléchargements partitionnés en cours sur un bucket. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple PHP suivant montre comment répertorier tous les chargements partitionnés en cours sur un compartiment.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);

// Write the list of uploads to the page.
print_r($result->toArray());
```

Liste des chargements partitionnés à l'aide de l'API REST

Les sections suivantes de la Référence de l'API Amazon Simple Storage Service décrivent l'API REST pour la liste des chargements partitionnés.

- ListParts-listez les parties téléchargées pour un téléchargement partitionné spécifique.
- ListMultipartUploads-liste les téléchargements partitionnés en cours.

Répertorier les téléchargements partitionnés à l'aide du AWS CLI

Les sections suivantes AWS Command Line Interface décrivent les opérations de listage des téléchargements partitionnés.

- list-parts- répertorie les parties chargées pour un chargement partitionné spécifique.
- list-multipart-uploads-liste les téléchargements partitionnés en cours.

Suivi d'un chargement partitionné

L'API de haut niveau de chargement partitionné fournit une interface d'écoute, `ProgressListener`, pour suivre la progression du chargement d'un objet sur Amazon S3. Les événements de progression ont lieu de manière périodique et informent l'écouteur que des octets ont été transférés.

Java

Example

```
TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

PutObjectRequest request = new PutObjectRequest(
    existingBucketName, keyName, new File(filePath));

// Subscribe to the event and provide event handler.
request.setProgressListener(new ProgressListener() {
   public void progressChanged(ProgressEvent event) {
    System.out.println("Transferred bytes: " +
      event.getBytesTransfered());
            }
});
```

Example

L'exemple de code Java suivant charge un fichier et utilise le `ProgressListener` pour suivre la progression du chargement. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import java.io.File;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.Upload;

public class TrackMPUProgressUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
```

```
        String existingBucketName = "*** Provide bucket name ***";
        String keyName            = "*** Provide object key ***";
        String filePath           = "*** file to upload ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        // For more advanced uploads, you can create a request object
        // and supply additional request parameters (ex: progress listeners,
        // canned ACLs, etc.)
        PutObjectRequest request = new PutObjectRequest(
          existingBucketName, keyName, new File(filePath));

        // You can ask the upload for its progress, or you can
        // add a ProgressListener to your request to receive notifications
        // when bytes are transferred.
        request.setGeneralProgressListener(new ProgressListener() {
   @Override
   public void progressChanged(ProgressEvent progressEvent) {
    System.out.println("Transferred bytes: " +
      progressEvent.getBytesTransferred());
   }
  });

        // TransferManager processes all transfers asynchronously,
        // so this call will return immediately.
        Upload upload = tm.upload(request);

        try {
         // You can block and wait for the upload to finish
         upload.waitForCompletion();
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload aborted.");
         amazonClientException.printStackTrace();
        }
    }
}
```

.NET

L'exemple C# suivant charge un fichier dans un compartiment S3 à l'aide de la classe `TransferUtility` et suit la progression du chargement. Pour plus d'informations sur la

configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object
 ***";
        private const string filePath = " *** provide the full path name of the file
 to upload **";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;


        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            TrackMPUAsync().Wait();
        }

        private static async Task TrackMPUAsync()
        {
            try
            {
                var fileTransferUtility = new TransferUtility(s3Client);

                // Use TransferUtilityUploadRequest to configure options.
                // In this example we subscribe to an event.
                var uploadRequest =
                    new TransferUtilityUploadRequest
                    {
                        BucketName = bucketName,
                        FilePath = filePath,
```

```
                        Key = keyName
                    };

                uploadRequest.UploadProgressEvent +=
                    new EventHandler<UploadProgressArgs>
                        (uploadRequest_UploadPartProgressEvent);

                await fileTransferUtility.UploadAsync(uploadRequest);
                Console.WriteLine("Upload completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }

        static void uploadRequest_UploadPartProgressEvent(object sender,
 UploadProgressArgs e)
        {
            // Process event.
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

Interruption d'un chargement partitionné

Après avoir lancé un chargement partitionné, vous commencez à charger les parties. Amazon S3 stocke ces parties, mais il crée l'objet à partir des parties uniquement après la fin de leur chargement et envoie une demande `successful` pour terminer le chargement partitionné (vous devez veiller à l'aboutissement de la demande pour terminer le chargement partitionné). Dès réception de la demande de fin de chargement partitionné, Amazon S3 assemble les parties et crée un objet. Si vous n'envoyez pas correctement la demande complète du chargement partitionné, Amazon S3 n'assemble pas les parties et ne crée aucun objet.

Vous êtes facturé pour tout le stockage associé aux parties chargées. Pour plus d'informations, consultez Chargement partitionné et tarification. Il est donc important de terminer le chargement partitionné pour que l'objet soit créé, ou d'arrêter le chargement partitionné pour supprimer les parties chargées.

Vous pouvez arrêter un chargement partitionné en cours dans Amazon S3 à l'aide de la AWS Command Line Interface (AWS CLI), de l'API REST ou AWS des kits SDK. Vous pouvez également arrêter un chargement partitionné incomplet à l'aide d'une configuration de cycle de vie de compartiment.

Utilisation des AWS SDK (API de haut niveau)

Java

La classe `TransferManager` fournit la méthode `abortMultipartUploads` pour arrêter les chargements partitionnés en cours. Un chargement est considéré comme étant en cours dès que vous le lancez et jusqu'à ce qu'il soit terminé ou arrêté. Vous fournissez une valeur `Date`, et cette API arrête tous les chargements partitionnés dans ce compartiment qui avaient été lancés avant la `Date` spécifiée et qui sont toujours en cours.

Les tâches suivantes vous guident tout au long de l'utilisation des classes Java de haut niveau pour arrêter les chargements partitionnés.

Processus d'arrêt des chargements partitionnés grâce à l'API de haut niveau

1	Créez une instance de la classe `TransferManager` .
2	Exécutez la méthode `TransferManager.abortMultipartUploads` en indiquant le nom du compartiment et une valeur `Date`.

Le code Java suivant arrête tous les chargements partitionnés en cours initiés dans un compartiment spécifique, il y a plus d'une semaine. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import java.util.Date;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3.transfer.TransferManager;

public class AbortMPUUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide existing bucket name ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        int sevenDays = 1000 * 60 * 60 * 24 * 7;
  Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);

        try {
         tm.abortMultipartUploads(existingBucketName, oneWeekAgo);
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload was aborted.");
         amazonClientException.printStackTrace();
        }
    }
}
```

> **Note**
>
> Vous pouvez également arrêter un chargement partitionné spécifique. Pour plus d'informations, consultez Utilisation des AWS SDK (API de bas niveau).

.NET

L'exemple C# suivant arrête tous les chargements partitionnés en cours initiés sur un compartiment spécifique il y a plus d'une semaine. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
```

```
{
    class AbortMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            AbortMPUAsync().Wait();
        }

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);

                // Abort all in-progress uploads initiated before the specified
 date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

> **Note**
>
> Vous pouvez également arrêter un chargement partitionné spécifique. Pour plus d'informations, consultez Utilisation des AWS SDK (API de bas niveau).

Utilisation des AWS SDK (API de bas niveau)

Vous pouvez arrêter un chargement partitionné en cours en appelant la méthode `AmazonS3.abortMultipartUpload`. Cette méthode supprime toutes les parties qui ont été chargées dans Amazon S3 et libère les ressources. Vous devez fournir l'ID de chargement, le nom du compartiment et le nom de la clé. L'exemple de code Java suivant illustre comment arrêter un chargement partitionné en cours.

Pour arrêter un chargement partitionné, vous fournissez l'ID de chargement, ainsi que les noms de compartiment et de clé utilisés dans le chargement. Après avoir arrêté un chargement partitionné, vous ne pouvez pas utiliser l'ID de chargement pour charger les parties supplémentaires. Pour en savoir plus sur les chargements partitionnés Amazon S3, veuillez consulter Chargement et copie d'objets à l'aide d'un chargement partitionné.

Java

L'exemple de code Java suivant arrête un chargement partitionné en cours.

Example

```
InitiateMultipartUploadRequest initRequest =
    new InitiateMultipartUploadRequest(existingBucketName, keyName);
InitiateMultipartUploadResult initResponse =
               s3Client.initiateMultipartUpload(initRequest);

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(
            existingBucketName, keyName, initResponse.getUploadId()));
```

> **Note**
>
> Au lieu d'un chargement partitionné spécifique, vous pouvez arrêter tous vos chargements partitionnés démarrés avant un moment spécifique et qui sont toujours en cours. Cette opération de nettoyage est utile pour arrêter les anciens chargements partitionnés que

vous avez initiés mais que vous n'avez pas terminés ou arrêtés. Pour plus d'informations, consultez Utilisation des AWS SDK (API de haut niveau).

.NET

L'exemple C# suivant montre comment arrêter un chargement partitionné. Pour obtenir un exemple C# complet incluant le code suivant, consultez Utilisation des AWS SDK (API de bas niveau).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Vous pouvez aussi annuler tous les chargements partitionnés en cours démarrés avant un moment spécifique. Cette opération de nettoyage est utile pour abandonner les chargements partitionnés que vous n'avez pas terminés ou interrompus. Pour plus d'informations, consultez Utilisation des AWS SDK (API de haut niveau).

PHP

Cet exemple montre comment utiliser une classe de la version 3 de AWS SDK for PHP pour abandonner un téléchargement partitionné en cours. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2. L'exemple de la méthode `abortMultipartUpload()`.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$uploadId = '*** Upload ID of upload to Abort ***';

$s3 = new S3Client([
```

```
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Abort the multipart upload.
$s3->abortMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
]);
```

Utilisation de l'API REST

Pour plus d'informations sur l'utilisation de l'API REST pour arrêter un téléchargement AbortMultipartUploaden plusieurs parties, consultez le manuel Amazon Simple Storage Service API Reference.

À l'aide du AWS CLI

Pour plus d'informations sur l'utilisation du AWS CLI pour arrêter un téléchargement partitionné, consultez abort-multipart-uploadla référence des AWS CLI commandes.

Copie d'un objet à l'aide du chargement partitionné

Les exemples de cette section vous montrent comment copier des objets supérieurs à 5 Go grâce à l'API de chargement partitionné. Vous pouvez copier des objets inférieurs à 5 Go en une seule opération. Pour plus d'informations, consultez Copier, déplacer et renommer des objets.

Utilisation des AWS SDK

Pour copier un objet à l'aide de l'API de bas niveau, procédez comme suit :

- Lancez un chargement partitionné en appelant la méthode `AmazonS3Client.initiateMultipartUpload()`.
- Enregistrez l'ID de chargement de l'objet de la réponse renvoyé par la méthode `AmazonS3Client.initiateMultipartUpload()`. Vous devez fournir cet ID de chargement pour chaque opération de chargement de partie.
- Copiez toutes les parties. Pour chaque partie que vous devez copier, créez une nouvelle instance de la classe `CopyPartRequest`. Fournissez les informations sur la partie, notamment les noms des compartiments source et de destination, les clés d'objet source et de destination, l'ID de

chargement, les emplacement des premier et dernier octets de la partie, ainsi que le numéro de la partie.

- Enregistrez les réponses des appels de méthode `AmazonS3Client.copyPart()`. Chaque réponse inclut la valeur `ETag` et le numéro de partie de la partie chargée. Vous avez besoin de ces informations pour terminer le chargement partitionné.
- Appelez la méthode `AmazonS3Client.completeMultipartUpload()` pour terminer l'opération de copie.

Java

Example

L'exemple suivant montre comment utiliser l'API Java de bas niveau Amazon S3 pour effectuer une copie en plusieurs parties. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartCopy {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String sourceBucketName = "*** Source bucket name ***";
        String sourceObjectKey = "*** Source object key ***";
        String destBucketName = "*** Target bucket name ***";
        String destObjectKey = "*** Target object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
```

```
                    .withRegion(clientRegion)
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(destBucketName,
                    destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make
sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize -
1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(sourceBucketName)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(destBucketName)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }
```

```
            // Complete the upload request to concatenate all uploaded parts and
 make the
            // copied object available.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest(
                    destBucketName,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

.NET

L'exemple C# suivant montre comment utiliser le AWS SDK for .NET pour copier un objet Amazon S3 d'une taille supérieure à 5 Go d'un emplacement source à un autre, par exemple d'un compartiment à un autre. Pour copier des objets dont la taille est inférieure à 5 Go, utilisez la procédure de copie en une seule opération décrite dans Utilisation des AWS SDK. Pour en savoir plus sur les chargements partitionnés Amazon S3, veuillez consulter Chargement et copie d'objets à l'aide d'un chargement partitionné.

Cet exemple montre comment copier un objet Amazon S3 d'une taille supérieure à 5 Go d'un compartiment S3 vers un autre à l'aide de l'API de téléchargement AWS SDK for .NET partitionné.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string targetBucket = "*** provide the name of the bucket to
 copy the object to ***";
        private const string sourceObjectKey = "*** provide the name of object to
 copy ***";
        private const string targetObjectKey = "*** provide the name of the object
 copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            MPUCopyObjectAsync().Wait();
        }
        private static async Task MPUCopyObjectAsync()
        {
            // Create a list to store the upload part responses.
            List<UploadPartResponse> uploadResponses = new
 List<UploadPartResponse>();
            List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();

            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest =
                new InitiateMultipartUploadRequest
```

```
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey
                };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Save the upload ID.
            String uploadId = initResponse.UploadId;

            try
            {
                // Get the size of the object.
                GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
                {
                    BucketName = sourceBucket,
                    Key = sourceObjectKey
                };

                GetObjectMetadataResponse metadataResponse =
                    await s3Client.GetObjectMetadataAsync(metadataRequest);
                long objectSize = metadataResponse.ContentLength; // Length in
bytes.

                // Copy the parts.
                long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

                long bytePosition = 0;
                for (int i = 1; bytePosition < objectSize; i++)
                {
                    CopyPartRequest copyRequest = new CopyPartRequest
                    {
                        DestinationBucket = targetBucket,
                        DestinationKey = targetObjectKey,
                        SourceBucket = sourceBucket,
                        SourceKey = sourceObjectKey,
                        UploadId = uploadId,
                        FirstByte = bytePosition,
                        LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
                        PartNumber = i
```

```
                    };

                    copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

                    bytePosition += partSize;
                }

                // Set up to complete the copy.
                CompleteMultipartUploadRequest completeRequest =
                new CompleteMultipartUploadRequest
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey,
                    UploadId = initResponse.UploadId
                };
                completeRequest.AddPartETags(copyResponses);

                // Complete the copy.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Utilisation de l'API REST

Les sections suivantes de la Référence de l'API Amazon Simple Storage Service décrivent l'API REST pour le chargement partitionné. Pour copier un objet existant, utilisez l'API de chargement d'une partie (Copy) et spécifiez l'objet source en ajoutant l'en-tête `x-amz-copy-source` dans votre demande.

- Lancement du chargement partitionné
- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Interruption du chargement partitionné
- Liste des parties
- Liste des chargements partitionnés

Vous pouvez utiliser ces API pour vos propres demandes REST ou vous pouvez utiliser l'un des kits SDK fournis. Pour plus d'informations sur l'utilisation du téléchargement partitionné avec AWS CLI le. À l'aide du AWS CLI Pour plus d'informations sur les kits SDK, consultez AWS Support du SDK pour le téléchargement en plusieurs parties.

Limites de la fonction de chargement partitionné Amazon S3

Le tableau suivant fournit les principales spécifications du chargement partitionné. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

Élément	Spécification
Taille maximum de l'objet	5 TiO
Nombre maximum de parties par chargement	10 000
Numéros de partie	1 à 10 000 (inclus)
Taille de partie	5 Mio à 5 GiO. Il n'existe pas de limite de taille minimale pour la dernière partie de votre chargement partitionné.
Nombre maximum de parties renvoyées pour une demande de liste des parties	1 000
Nombre maximum de chargements partitionnés renvoyés dans une	1 000

Élément	Spécification
demande de liste de chargements partitionnés	

Copier, déplacer et renommer des objets

L'`CopyObject`opération crée une copie d'un objet déjà stocké dans Amazon S3.

Vous pouvez créer une copie d'un objet d'une capacité maximale de 5 Go en une seule opération atomique. Toutefois, pour copier un objet dont la taille est supérieure à 5 Go, vous devez utiliser un téléchargement partitionné. Pour plus d'informations, consultez the section called “Copier un objet”.

Grâce à l'opération `CopyObject`, vous pouvez :

- Créez des copies supplémentaires d'objets.
- Renommez les objets en les copiant et en supprimant les objets d'origine.
- Copiez ou déplacez des objets d'un compartiment à un autre, y compris entre eux Régions AWS (par exemple, de `us-west-1` à`eu-west-2`). Lorsque vous déplacez un objet, Amazon S3 copie l'objet vers la destination spécifiée, puis supprime l'objet source.

  > **Note**
  >
  > Copier ou déplacer des objets Régions AWS entraîne des frais de bande passante. Pour plus d'informations, consultez Tarification Amazon S3.

- Modifiez les métadonnées de l'objet. Chaque objet Amazon S3 possède des métadonnées. Ces métadonnées sont un ensemble de paires nom-valeur. Vous pouvez définir les métadonnées d'un objet au moment où vous le chargez. Une fois que vous avez chargé l'objet, vous ne pouvez pas modifier les métadonnées de l'objet. Le seul moyen de modifier les métadonnées d'objet est de faire une copie de l'objet et de configurer les métadonnées. Pour ce faire, lors de l'opération de copie, définissez le même objet que la source et la cible.

  Certaines métadonnées d'objet sont des métadonnées système, tandis que d'autres sont définies par l'utilisateur. Vous pouvez contrôler certaines métadonnées du système. Par exemple, vous pouvez contrôler la classe de stockage et le type de chiffrement côté serveur à utiliser pour l'objet. Lorsque vous copiez un objet, les métadonnées système contrôlées par l'utilisateur et

les métadonnées définies par l'utilisateur sont également copiées. Amazon S3 réinitialise les métadonnées contrôlées par le système. Par exemple, lorsque vous copiez un objet, Amazon S3 réinitialise la date de création de l'objet copié. Il n'est pas nécessaire de définir ces valeurs de métadonnées contrôlées par le système dans votre demande de copie.

Lorsque vous copiez un objet, vous devez choisir de mettre à jour certaines valeurs des métadonnées. Par exemple, si votre objet source est configuré pour utiliser le stockage S3 Standard, vous pouvez sélectionner S3 Intelligent-Tiering pour la copie de l'objet. Vous devez également choisir de modifier certaines valeurs des métadonnées définies par l'utilisateur présentes dans l'objet source. Si vous choisissez de mettre à jour les métadonnées (système ou définies par l'utilisateur) configurables par l'utilisateur de l'objet lors de la copie, vous devez explicitement spécifier toutes les métadonnées configurables par l'utilisateur présentes dans l'objet source de la demande, même si vous ne changez qu'une des valeurs des métadonnées.

Pour en savoir plus sur les métadonnées d'objet, consultez Utilisation des métadonnées d'objet.

Copie d'objets archivés et restaurés

Si l'objet source est archivé dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, vous devez d'abord restaurer une copie temporaire avant de copier l'objet vers un autre compartiment. Pour en savoir plus sur l'archivage des objets, consultez Transition vers les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive (archivage d'objets).

L'opération de copie dans la console Amazon S3 n'est pas prise en charge pour les objets restaurés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Pour copier ces objets restaurés, utilisez le AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST Amazon S3.

Copier des objets chiffrés

Amazon S3 chiffre automatiquement tous les nouveaux objets copiés dans un compartiment S3. Si vous ne spécifiez aucune information de chiffrement dans votre demande de copie, le paramètre de chiffrement de l'objet cible est défini sur la configuration de chiffrement par défaut du compartiment de destination. Par défaut, tous les compartiments ont un niveau de chiffrement de base qui utilise le chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3). Si le compartiment de destination possède une configuration de chiffrement par défaut qui utilise le chiffrement côté serveur avec une clé AWS Key Management Service (AWS KMS) (SSE-KMS) ou une clé de chiffrement fournie par le client (SSE-C), Amazon S3 utilise la clé KMS correspondante, ou une clé fournie par le client pour chiffrer la copie de l'objet cible.

Lorsque vous copiez un objet, si vous souhaitez utiliser un autre type de paramètre de chiffrement pour l'objet cible, vous pouvez demander à Amazon S3 de chiffrer l'objet cible à l'aide d'une clé KMS, d'une clé gérée par Amazon S3 ou d'une clé fournie par le client. Si le paramètre de chiffrement de votre demande est différent de la configuration de chiffrement par défaut du compartiment de destination, le paramètre de chiffrement de votre demande est prioritaire. Si l'objet source de la copie est chiffré avec SSE-C, vous devez fournir les informations de chiffrement nécessaires dans votre demande afin qu'Amazon S3 puisse déchiffrer l'objet à copier. Pour plus d'informations, consultez Protection des données à l'aide du chiffrement.

Utilisation de checksums lors de la copie d'objets

Lorsque vous copiez des objets, vous pouvez sélectionner un algorithme de total de contrôle différent pour l'objet. Que vous sélectionniez le même algorithme ou un nouvel algorithme, Amazon S3 calcule un nouveau total de contrôle après la copie de l'objet. Amazon S3 ne copie pas directement la valeur du total de contrôle. La valeur de la somme de contrôle des objets chargés à l'aide de téléchargements partitionnés peut changer. Pour plus d'informations sur le mode de calcul du total de contrôle, consultez la section Utilisation de totaux de contrôle au niveau des parties pour les chargements partitionnés.

Copier plusieurs objets en une seule demande

Pour copier plusieurs objets Amazon S3 avec une seule demande, vous pouvez également utiliser S3 Batch Operations. Vous fournissez à la fonctionnalité d'opérations par lot S3 une liste d'objets sur lesquels effectuer des opérations. La fonctionnalité des opérations par lot S3 appelle l'opération d'API respective pour effectuer l'opération spécifiée. Une tâche d'opérations par lot peut effectuer l'opération spécifiée sur des milliards d'objets contenant des exaoctets de données.

La fonctionnalité d'opérations par lot S3 suit la progression, envoie des notifications et stocke un rapport de fin détaillé sur toutes les actions, offrant ainsi une expérience sans serveur entièrement gérée et qui peut être vérifiée. Vous pouvez utiliser S3 Batch Operations via la console Amazon S3 AWS CLI, AWS les SDK ou l'API REST. Pour plus d'informations, consultez the section called “Principes de base des opérations par lot”.

Copier des objets dans des compartiments de répertoire

Pour plus d'informations sur la copie d'un objet dans un compartiment de répertoire, consultezCopie d’un objet vers un compartiment de répertoires. Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoire, consultez Qu’est-ce que S3 Express One Zone ? etCompartiments de répertoire.

Pour copier un objet

Pour copier un objet, utilisez les méthodes suivantes.

Utilisation de la console S3

> **Note**
>
> - Lorsque vous copiez un objet à l'aide de la console Amazon S3, vous devez en avoir l'`s3:ListAllMyBuckets`autorisation. La console a besoin de cette autorisation pour valider l'opération de copie. Pour des exemples de politiques qui accordent cette autorisation, voirthe section called “Exemples de politiques basées sur l’identité”.
>
> Si vous copiez un objet doté de balises définies par l'utilisateur, vous devez également en avoir l'`s3:GetObjectTagging`autorisation. Si vous copiez un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`GetObjectTagging`autorisation s3 :.
>
> Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, l'objet sera copié sans les balises définies par l'utilisateur et vous recevrez un message d'erreur.
> - Les objets chiffrés à l'aide des clés de chiffrement fournies par le client (SSE-C) ne peuvent pas être copiés à l'aide de la console S3. Pour copier des objets chiffrés avec SSE-C, utilisez le AWS CLI AWS SDK ou l'API REST Amazon S3.
> - La copie interrégionale d'objets chiffrés avec SSE-KMS n'est pas prise en charge par la console Amazon S3. Pour copier des objets chiffrés avec SSE-KMS d'une région à l'autre AWS CLI, utilisez le AWS SDK ou l'API REST Amazon S3.

Pour copier un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Buckets, puis choisissez l'onglet Buckets à usage général. Accédez au compartiment ou au dossier Amazon S3 contenant les objets que vous souhaitez copier.
3. Cochez la case située à gauche des noms des objets que vous souhaitez copier.

4. Dans le menu Actions, choisissez Copier dans la liste des options qui s'affiche.
5. Sélectionnez le type de destination et le compte de destination. Pour spécifier le chemin de destination, choisissez Browse S3 (Parcourir S3), accédez à la destination et activez la case à cocher située à gauche de la destination. Choisissez Choose destination (Choisir une destination) en bas à droite.

   Vous pouvez également saisir le chemin de destination.
6. Si la gestion des versions de compartiment n'est pas activée, il se peut que vous soyez invité à reconnaître que les objets existants portant le même nom soient remplacés. Si cela vous convient, activez la case à cocher et continuez. Si vous souhaitez conserver toutes les versions des objets dans ce compartiment, sélectionnez Enable Bucket Versioning (Activer la gestion des versions pour le compartiment). Vous pouvez également mettre à jour les propriétés de chiffrement et de verrouillage des objets S3 par défaut.
7. Sous Additional checksums (Totaux de contrôle supplémentaires), sélectionnez si vous souhaitez copier les objets à l'aide de la fonction de total de contrôle existante ou remplacer la fonction de total de contrôle existante par une nouvelle. Lorsque vous avez chargé les objets, vous aviez la possibilité de spécifier l'algorithme total de contrôle utilisé pour vérifier l'intégrité des données. Lors de la copie de l'objet, vous avez la possibilité de sélectionner une nouvelle fonction. Si vous n'avez pas spécifié de total de contrôle supplémentaire à l'origine, vous pouvez utiliser la section des options de copie pour en ajouter un.

   > **Note**
   >
   > Même si vous choisissez d'utiliser la même fonction de total de contrôle, la valeur de votre total de contrôle peut changer si vous copiez l'objet et que sa taille dépasse 16 Mo. La valeur du total de contrôle peut changer en raison de la façon dont les totaux de contrôle sont calculés pour les chargements partitionnés. Pour plus d'informations sur la façon dont le total de contrôle peut changer lors de la copie de l'objet, consultez Utilisation de totaux de contrôle au niveau des parties pour les chargements partitionnés.

   Pour modifier la fonction de total de contrôle, sélectionnez Replace with a new checksum function (Remplacer par une nouvelle fonction de total de contrôle). Sélectionnez la nouvelle fonction de total de contrôle dans la case. Lorsque l'objet est copié, le nouveau total de contrôle est calculé et stocké en utilisant l'algorithme spécifié.
8. En bas à droite, choisissez Copy (Copier). Amazon S3 copie votre objet dans la destination.

Utilisation des AWS SDK

Les exemples de cette section vous montrent comment copier des objets supérieurs à 5 Go en une seule opération. Pour copier des objets de plus de 5 Go, vous devez utiliser un téléchargement partitionné. Pour plus d'informations, consultez Copie d'un objet à l'aide du chargement partitionné.

Java

Example

L'exemple suivant copie un objet dans Amazon S3 à l'aide du kit AWS SDK for Java. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

import java.io.IOException;

public class CopyObjectSingleOperation {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String sourceKey = "*** Source object key *** ";
        String destinationKey = "*** Destination object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName,
 sourceKey, bucketName, destinationKey);
            s3Client.copyObject(copyObjRequest);
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

L'exemple C# suivant utilise le niveau élevé AWS SDK for .NET pour copier des objets d'une taille maximale de 5 Go en une seule opération. Pour les objets de taille supérieurs à 5 Go, utilisez l'exemple de copie de chargement partitionné décrit dans Copie d'un objet à l'aide du chargement partitionné.

Cet exemple effectue la copie d'un objet de 5 Go au plus. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string destinationBucket = "*** provide the name of the bucket
 to copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy
 ***";
        private const string destObjectKey = "*** provide the destination object key
 name ***";
        // Specify your bucket region (an example region is shown).
```

```
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            CopyingObjectAsync().Wait();
        }

        private static async Task CopyingObjectAsync()
        {
            try
            {
                CopyObjectRequest request = new CopyObjectRequest
                {
                    SourceBucket = sourceBucket,
                    SourceKey = objectKey,
                    DestinationBucket = destinationBucket,
                    DestinationKey = destObjectKey
                };
                CopyObjectResponse response = await
 s3Client.CopyObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Cette rubrique explique comment utiliser les classes de la version 3 de AWS SDK for PHP pour copier un seul objet et plusieurs objets au sein d'Amazon S3, d'un compartiment à un autre ou au sein du même compartiment.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple PHP suivant illustre l'utilisation de `copyObject()` cette méthode pour copier un seul objet dans Amazon S3. Il montre également comment créer plusieurs copies d'un objet en utilisant un lot d'appels `CopyObject` à la `getcommand()` méthode.

Copie d'objets

1	Créez l'instance d'un client Simple Storage Service (Amazon S3) à l'aide du constructeur de classe `Aws\S3\S3Client` .
2	Pour créer plusieurs copies d'un objet, vous exécutez un lot d'appels à la getCommand()méthode client Amazon S3, héritée de la Aws\CommandInterfaceclasse. Vous fournissez la commande `CopyObject` comme le premier argument et un tableau contenant le compartiment source, le nom de la clé source, le compartiment cible et le nom de la clé cible comme second argument.

```
 require 'vendor/autoload.php';

use Aws\CommandPool;
use Aws\Exception\AwsException;
use Aws\ResultInterface;
use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Copy an object.
$s3->copyObject([
```

```
    'Bucket' => $targetBucket,
    'Key' => "$sourceKeyname-copy",
    'CopySource' => "$sourceBucket/$sourceKeyname",
]);

// Perform a batch of CopyObject operations.
$batch = array();
for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket' => $targetBucket,
        'Key' => "{targetKeyname}-$i",
        'CopySource' => "$sourceBucket/$sourceKeyname",
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach ($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (Exception $e) {
    // General error handling here
}
```

Python

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
 Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key
```

```
    def copy(self, dest_object):
        """
        Copies the object to another bucket.

        :param dest_object: The destination object initialized with a bucket and
key.
                            This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
                CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
            )
            dest_object.wait_until_exists()
            logger.info(
                "Copied object from %s:%s to %s:%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
        except ClientError:
            logger.exception(
                "Couldn't copy object from %s/%s to %s/%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
            raise
```

Ruby

Les tâches suivantes vous guident dans l'utilisation Ruby des classes pour copier un objet dans Amazon S3 d'un compartiment à un autre ou au sein du même compartiment.

Copie d'objets

1	Utilisez la gemme modulaire Amazon S3 pour la version 3 de AWS SDK for Rubyaws-sdk-s3 , exigez et fournissez vos AWS informations d'identification. Pour plus d'informations sur la manière de fournir vos informations d'identification,

	consultez Faire des demandes à l'aide des Compte AWS informations d'identification utilisateur ou IAM.
2	Fournissez les informations de demande, telles que le nom du compartiment source, le nom de la clé source, le nom du compartiment de destination et la clé de destination.

L'exemple de Ruby code suivant illustre les tâches précédentes en utilisant la `#copy_object` méthode pour copier un objet d'un compartiment vers un autre.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket and rename it with the
 target key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Example usage:
def run_demo
```

```
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Utilisation de l'API REST

Cet exemple décrit comment copier un objet à l'aide de l'API REST Amazon S3. Pour plus d'informations sur l'API REST, consultez CopyObject.

L'exemple copie l'objet `flotsam` du compartiment *`example-s3-bucket1`* vers l'objet `jetsam` du compartiment *`example-s3-bucket2`*, en préservant les métadonnées.

```
PUT /jetsam HTTP/1.1
Host: example-s3-bucket2.s3.amazonaws.com
x-amz-copy-source: /example-s3-bucket1/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

La signature est générée à partir des informations suivantes.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/example-s3-bucket1/flotsam\r\n
/example-s3-bucket2/jetsam
```

Amazon S3 renvoie la réponse suivante qui spécifie l'ETag de l'objet et la date de modification.

```
HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbzv34BnSu5hctyyNSlHTYZFMWK4FtzO+iX8JQNyaLdTshL0KxatbaOZt
x-amz-request-id: 6B13C3C5B34AF333
Date: Wed, 20 Feb 2008 22:13:01 +0000

Content-Type: application/xml
Transfer-Encoding: chunked
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
   <LastModified>2008-02-20T22:13:01</LastModified>
   <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

À l'aide du AWS CLI

Vous pouvez également utiliser le AWS Command Line Interface (AWS CLI) pour copier un objet S3. Pour plus d'informations, consultez la section copy-object dans la référence des commandes AWS CLI .

Pour plus d'informations sur le AWS CLI, voir Qu'est-ce que le AWS Command Line Interface ? dans le guide de AWS Command Line Interface l'utilisateur.

Pour déplacer un objet

Pour déplacer un objet, appliquez les méthodes suivantes.

Utilisation de la console S3

> **Note**
>
> - Si vous déplacez un objet doté de balises définies par l'utilisateur, vous devez en avoir l'`s3:GetObjectTagging`autorisation. Si vous déplacez un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`GetObjectTagging`autorisation s3 :.

Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, l'objet sera déplacé sans les balises définies par l'utilisateur et vous recevrez un message d'erreur.

- Les objets chiffrés à l'aide des clés de chiffrement fournies par le client (SSE-C) ne peuvent pas être déplacés à l'aide de la console Amazon S3. Pour déplacer des objets chiffrés avec SSE-C, utilisez les AWS CLI AWS SDK ou l'API REST Amazon S3.
- Lorsque vous déplacez des dossiers, attendez que l'opération de déplacement soit terminée avant d'apporter des modifications supplémentaires aux dossiers.
- Vous ne pouvez pas utiliser les alias de point d'accès S3 comme source ou destination pour les opérations Move dans la console Amazon S3.

Pour déplacer un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Buckets, puis choisissez l'onglet Buckets à usage général. Accédez au compartiment ou au dossier Amazon S3 contenant les objets que vous souhaitez déplacer.
3. Activez la case à cocher située à gauche des noms des objets à déplacer.
4. Dans le menu Actions, choisissez Déplacer.
5. Pour spécifier le chemin de destination, choisissez Browse S3 (Parcourir S3), accédez à la destination et activez la case à cocher située à gauche de la destination. Choisissez Choose destination (Choisir une destination) en bas à droite.

   Vous pouvez également saisir le chemin de destination.
6. Si la gestion des versions de compartiment n'est pas activée, il se peut que vous soyez invité à reconnaître que les objets existants portant le même nom soient remplacés. Si cela vous convient, activez la case à cocher et continuez. Si vous souhaitez conserver toutes les versions des objets dans ce compartiment, sélectionnez Enable Bucket Versioning (Activer la gestion des versions pour le compartiment). Vous pouvez également mettre à jour les propriétés de chiffrement et de verrouillage des objets par défaut.
7. En bas à droite, choisissez Move (Déplacer). Amazon S3 déplace vos objets vers la destination.

> **Note**
>
> - Cette action crée une copie de tous les objets spécifiés avec des paramètres mis à jour, met à jour la date de dernière modification à l'emplacement spécifié et ajoute un marqueur de suppression à l'objet d'origine.
> - Cette action met à jour les métadonnées pour le contrôle de version du compartiment, le chiffrement, les fonctions de verrouillage des objets et les objets archivés.

À l'aide du AWS CLI

Vous pouvez également utiliser le AWS Command Line Interface (AWS CLI) pour déplacer un objet S3. Pour plus d'informations, consultez la section mv dans la référence des commandes AWS CLI .

Pour plus d'informations sur le AWS CLI, voir Qu'est-ce que le AWS Command Line Interface ? dans le guide de AWS Command Line Interface l'utilisateur.

Pour renommer un objet

Pour renommer un objet, procédez comme suit.

> **Note**
>
> - Le changement de nom d'un objet crée une copie de l'objet avec une nouvelle date de dernière modification, puis ajoute un marqueur de suppression à l'objet d'origine.
> - Les paramètres du compartiment pour le chiffrement par défaut sont automatiquement appliqués à tout objet spécifié non chiffré.
> - Vous ne pouvez pas utiliser la console Amazon S3 pour renommer des objets à l'aide de clés de chiffrement fournies par le client (SSE-C). Pour renommer des objets chiffrés avec SSE-C, utilisez les AWS SDK AWS CLI ou l'API REST Amazon S3 pour copier ces objets sous un nouveau nom.
> - Si ce compartiment utilise le paramètre imposé par le propriétaire du compartiment pour la propriété des objets S3, les listes de contrôle d'accès aux objets (ACL) ne seront pas copiées.
> - Si vous renommez un objet dont les balises sont définies par l'utilisateur, vous devez en avoir l'`s3:GetObjectTagging`autorisation. Si vous renommez un objet qui ne possède

pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'autorisation s3 :`GetObjectTagging`.

Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, l'objet sera renommé, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

Pour renommer un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Buckets, puis choisissez l'onglet Buckets à usage général. Accédez au compartiment ou au dossier Amazon S3 contenant l'objet que vous souhaitez renommer.
3. Cochez la case située à gauche du nom de l'objet que vous souhaitez renommer.
4. Dans le menu Actions, choisissez Renommer l'objet.
5. Dans le champ Nom du nouvel objet, entrez le nouveau nom de l'objet.
6. Choisissez Enregistrer les modifications dans le coin inférieur droit. Amazon S3 renomme votre objet.

Téléchargement d'objets

Cette section explique comment télécharger des objets depuis un compartiment Amazon S3. Avec Amazon S3, vous pouvez stocker des objets dans un ou plusieurs compartiments, et chaque objet peut atteindre une taille de 5 To. Tout objet Amazon S3 non archivé est accessible en temps réel. Les objets archivés doivent toutefois être restaurés avant de pouvoir être téléchargés. Pour obtenir des informations sur le téléchargement des objets archivés, consultez the section called “Téléchargement des objets archivés”.

Vous pouvez télécharger un seul objet à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI), AWS des SDK ou de l'API REST Amazon S3. Pour télécharger un objet depuis S3 sans écrire de code ni exécuter aucune commande, utilisez la console S3. Pour plus d’informations, consultez the section called “Téléchargement d'un objet”.

Pour télécharger plusieurs objets AWS CloudShell, utilisez le AWS CLI ou les AWS SDK. Pour plus d’informations, consultez the section called “Téléchargement de plusieurs objets”.

Si vous devez télécharger une partie d'un objet, vous devez utiliser des paramètres supplémentaires avec l'API REST AWS CLI ou pour spécifier uniquement les octets que vous souhaitez télécharger. Pour plus d'informations, consultez the section called “Téléchargement d'une partie d'un objet”.

Si vous devez télécharger un objet qui ne vous appartient pas, demandez à son propriétaire de générer une URL présignée qui vous permettra de télécharger l'objet. Pour plus d'informations, consultez the section called “Téléchargement d'un objet à partir d'un autre Compte AWS”.

Lorsque vous téléchargez des objets en dehors du AWS réseau, des frais de transfert de données s'appliquent. Le transfert de données au sein du AWS réseau est gratuit Région AWS, mais toute `GET` demande vous sera facturée. Pour plus d'informations sur les coûts de transfert de données et les frais de récupération de données, consultez Tarification Amazon S3.

Rubriques

Téléchargement d'un objet

Vous pouvez télécharger un objet à l'aide de la console Amazon S3 AWS CLI, des AWS kits SDK ou de l'API REST.

Utilisation de la console S3

Cette section explique comment utiliser la console Amazon S3 pour télécharger un objet depuis un compartiment S3.

> **Note**
>
> - Vous ne pouvez télécharger qu'un seul objet à la fois.
> - Si vous utilisez la console Amazon S3 pour télécharger un objet dont le nom de clé se termine par un point (`.`), celui-ci est supprimé du nom de clé de l'objet téléchargé.

Pour conserver le délai à la fin du nom de l'objet téléchargé, vous devez utiliser le AWS Command Line Interface (AWS CLI), AWS les SDK ou l'API REST Amazon S3.

Pour télécharger un objet à partir d'un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment à partir duquel vous souhaitez télécharger un objet.
3. Vous pouvez télécharger un objet à partir d'un compartiment S3 de l'une des façons suivantes :
   - Cochez la case à côté de l'objet et choisissez Télécharger. Si vous souhaitez télécharger l'objet dans un dossier spécifique, dans le menu Actions, choisissez Télécharger en tant que.
   - Si vous souhaitez télécharger une version spécifique de l'objet, activez Afficher les versions (en regard de la zone de recherche). Cochez la case à côté de la version de l'objet de votre choix, puis choisissez Télécharger. Si vous souhaitez télécharger l'objet dans un dossier spécifique, dans le menu Actions, choisissez Télécharger en tant que.

À l'aide du AWS CLI

L'exemple de commande `get-object` suivant montre comment utiliser l'interface AWS CLI pour télécharger un objet depuis Amazon S3. Cette commande récupère l'objet *`folder/my_image`* à partir du compartiment *`example-s3-bucket1`*. L'objet sera téléchargé dans un fichier nommé *`my_downloaded_image`*.

```
aws s3api get-object --bucket example-s3-bucket1 --key folder/
my_image my_downloaded_image
```

Pour plus d'informations et des exemples, consultez get-object dans la Référence des commandes AWS CLI .

Utilisation des AWS SDK

Pour des exemples de téléchargement d'un objet à l'aide AWS des SDK, consultezUtilisation `GetObject` avec un AWS SDK ou une CLI.

Pour des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour récupérer des objets depuis Amazon S3. Pour plus d'informations, veuillez consulter GetObject dans la Référence d'API Amazon Simple Storage Service.

Téléchargement de plusieurs objets

Vous pouvez télécharger plusieurs objets en utilisant AWS CloudShell AWS CLI le ou les AWS SDK.

En utilisant AWS CloudShell dans le AWS Management Console

AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console

Pour plus d'informations AWS CloudShell, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

> **Important**
>
> Avec AWS CloudShell, votre répertoire personnel dispose d'un espace de stockage allant jusqu'à 1 Go par. Région AWS Par conséquent, vous ne pouvez pas synchroniser les compartiments avec des objets dont le total est supérieur à ce montant. Pour plus de restrictions, consultez Quotas et restrictions des services dans le Guide de l'utilisateur AWS CloudShell .

Pour télécharger des objets en utilisant AWS CloudShell

1. Connectez-vous à la CloudShell console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudshell/.
2. Exécutez la commande suivante pour synchroniser les objets de votre bucket avec CloudShell. La commande suivante synchronise les objets du compartiment nommé *example-s3-bucket1* et crée un dossier nommé *temp* dans CloudShell. CloudShell synchronise vos objets dans ce dossier. Pour utiliser cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3 sync s3://example-s3-bucket1 ./temp
```

> **Note**
>
> Pour effectuer une correspondance de modèles afin d'exclure ou d'inclure des objets particuliers, vous pouvez utiliser les paramètres `--exclude "value"` et `--include "value"` avec la commande `sync`.

3. Exécutez la commande suivante pour compresser vos objets dans le dossier nommé *temp* dans un fichier nommé *temp.zip*.

```
zip temp.zip -r temp/
```

4. Choisissez Actions, puis choisissez Télécharger un fichier.
5. Entrez le nom de fichier **temp.zip**, puis choisissez Télécharger.
6. (Facultatif) Supprimez le *temp.zip* fichier et les objets synchronisés avec le *temp* dossier dans CloudShell. Avec AWS CloudShell, vous disposez d'un stockage persistant allant jusqu'à 1 Go pour chaque Région AWS.

   Vous pouvez utiliser l'exemple de commande suivant pour supprimer votre fichier `.zip` et votre dossier. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
rm temp.zip && rm -rf temp/
```

À l'aide du AWS CLI

L'exemple suivant montre comment vous pouvez utiliser le AWS CLI pour télécharger tous les fichiers ou objets du répertoire ou du préfixe spécifié. Cette commande copie tous les objets du compartiment *example-s3-bucket1* vers votre répertoire actuel. Pour utiliser cet exemple de commande, utilisez le nom de votre compartiment à la place de *example-s3-bucket1*.

```
aws s3 cp s3://example-s3-bucket1 . --recursive
```

La commande suivante télécharge tous les objets situés sous le préfixe *logs* du compartiment *example-s3-bucket1* dans votre répertoire actuel. Elle utilise également les paramètres `--`

`exclude` et `--include` pour copier uniquement les objets dotés du suffixe *`.log`*. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3 cp s3://example-s3-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Pour plus d'informations et des exemples, consultez cp dans la Référence des commandes AWS CLI .

Utilisation des AWS SDK

Pour des exemples de téléchargement de tous les objets d'un compartiment Amazon S3 avec les AWS SDK, consultezTéléchargez tous les objets d'un compartiment Amazon Simple Storage Service (Amazon S3) dans un répertoire local..

Pour des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Téléchargement d'une partie d'un objet

Vous pouvez télécharger une partie d'un objet à l'aide de l' AWS CLI API REST. Pour ce faire, vous devez utiliser des paramètres supplémentaires pour spécifier la partie d'un objet que vous souhaitez télécharger.

À l'aide du AWS CLI

L'exemple de commande suivant exécute une demande `GET` pour obtenir une plage d'octets dans l'objet nommé *`folder/my_data`* dans le compartiment nommé *`example-s3-bucket1`*. Dans cette demande, la plage d'octets doit être préfixée par `bytes=`. L'objet partiel est téléchargé dans le fichier de sortie nommé *`my_data_range`*. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api get-object --bucket example-s3-bucket1 --key folder/my_data --range
 bytes=0-500 my_data_range
```

Pour plus d'informations et des exemples, consultez get-object dans la Référence des commandes AWS CLI .

Pour plus d'informations sur l'en-tête HTTP `Range`, consultez RFC 9110 sur le site web RFC Editor.

> **Note**
>
> Amazon S3 ne prend pas en charge la récupération de plusieurs plages de données dans une demande GET individuelle.

Utilisation de l'API REST

Vous pouvez utiliser les paramètres `partNumber` et `Range` de l'API REST pour récupérer des parties d'objet depuis Amazon S3. Pour plus d'informations, veuillez consulter GetObject dans la Référence d'API Amazon Simple Storage Service.

Téléchargement d'un objet à partir d'un autre Compte AWS

Vous pouvez utiliser une URL présignée pour accorder aux autres un accès limité dans le temps à vos objets sans mettre à jour votre stratégie de compartiment.

L'URL présignée peut être saisie dans un navigateur ou utilisée par un programme pour télécharger un objet. Les informations d'identification utilisées par l'URL sont celles de l' AWS utilisateur qui a généré l'URL. Une fois l'URL créée, toute personne disposant de l'URL présignée peut télécharger l'objet correspondant jusqu'à ce que l'URL expire.

Utilisation d'une URL présignée dans la console S3

Vous pouvez utiliser la console Amazon S3 afin de générer une URL présignée pour partager un objet en suivant ces étapes. Lors de l'utilisation de la console, le délai d'expiration maximal d'une URL présignée est de 12 heures à compter de la création.

Pour générer une URL présignée à l'aide de la console Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet pour lequel vous souhaitez une URL pré-signée.
4. Dans la liste Objects (Objets), sélectionnez l'objet pour lequel vous souhaitez créer une URL présignée.
5. Dans le menu Actions d'objet, choisissez Partager avec une URL présignée.
6. Spécifiez la durée de validité souhaitée pour l'URL présignée.

7. Choisissez Create presigned URL (Créer une URL présignée).
8. Lorsqu'un message de confirmation apparaît, l'URL est automatiquement copiée dans votre presse-papier. Un bouton s'affiche pour copier l'URL présignée, si vous devez la copier à nouveau.
9. Pour télécharger l'objet, collez l'URL dans n'importe quel navigateur et l'objet tentera de se télécharger.

Pour plus d'informations sur les URL présignées et les autres méthodes permettant de les créer, consultez Utilisation d'URL présignées.

Téléchargement des objets archivés

Pour réduire vos coûts de stockage pour les objets rarement consultés, vous pouvez archiver ces objets. Lorsque vous archivez un objet, il est placé dans un espace de stockage à faible coût, ce qui signifie que vous ne pouvez pas y accéder en temps réel. Pour télécharger un objet archivé, vous devez d'abord le restaurer.

Vous pouvez restaurer les objets archivés en quelques minutes ou quelques heures, selon la classe de stockage. Vous pouvez restaurer un objet archivé à l'aide de la console Amazon S3, de S3 Batch Operations, de l'API REST Amazon S3, AWS des SDK et du AWS Command Line Interface (AWS CLI).

Pour obtenir des instructions, veuillez consulter Restauration d'un objet archivé. Après avoir restauré l'objet archivé, vous pouvez le télécharger.

Résolution des problèmes de téléchargement d'objets

Des autorisations insuffisantes ou des politiques de bucket ou d'utilisateur AWS Identity and Access Management (IAM) incorrectes peuvent provoquer des erreurs lorsque vous essayez de télécharger des objets depuis Amazon S3. Ces problèmes peuvent souvent provoquer des erreurs d'accès refusé (403 Interdit), quand Amazon S3 ne parvient pas à autoriser l'accès à une ressource.

Pour connaître les causes courantes des erreurs d'accès refusé (403 – Interdit), consultez Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3.

Vérification de l'intégrité des objets

Amazon S3 utilise des valeurs de total de contrôle pour vérifier l'intégrité des données que vous chargez ou téléchargez depuis Amazon S3. En outre, vous pouvez demander qu'un autre total de

contrôle soit calculé pour tout objet que vous stockez dans Amazon S3. Vous pouvez choisir parmi plusieurs algorithmes de total de contrôle à utiliser lorsque vous chargez ou copiez vos données. Amazon S3 utilise cet algorithme pour calculer un total de contrôle supplémentaire et le stocker dans les métadonnées de l'objet. Pour en savoir plus sur l'utilisation de totaux de contrôle supplémentaires pour vérifier l'intégrité des données, consultez Tutoriel : vérifier l'intégrité des données dans Amazon S3 avec des totaux de contrôle supplémentaires.

Lorsque vous chargez un objet, vous pouvez éventuellement inclure un total de contrôle précalculé dans votre requête. Amazon S3 compare le total de contrôle fourni au total de contrôle qu'il calcule en utilisant l'algorithme que vous avez spécifié. Si les deux valeurs ne correspondent pas, Amazon S3 génère une erreur.

Utilisation des algorithmes de total de contrôle pris en charge

Amazon S3 vous offre la possibilité de sélectionner l'algorithme de total de contrôle utilisé pour valider vos données pendant le chargement ou le téléchargement. Vous pouvez sélectionner l'un des algorithmes de contrôle d'intégrité des données suivants : Secure Hash Algorithms (SHA) ou Cyclic Redundancy Check (CRC) :

- CRC32
- CRC32C
- SHA-1
- SHA-256

Lorsque vous chargez un objet, vous pouvez spécifier l'algorithme que vous souhaitez utiliser :

- Lorsque vous utilisez le AWS Management Console, vous sélectionnez l'algorithme de somme de contrôle que vous souhaitez utiliser. Dans ce cas, vous pouvez éventuellement spécifier la valeur du total de contrôle de l'objet. Lorsque Amazon S3 reçoit l'objet, il calcule le total de contrôle en utilisant l'algorithme que vous avez spécifié. Si les deux valeurs ne correspondent pas, Amazon S3 génère une erreur.
- Lorsque vous utilisez un kit SDK, vous pouvez définir la valeur du paramètre `x-amz-sdk-checksum-algorithm` sur l'algorithme que vous souhaitez que Amazon S3 utilise pour calculer le total de contrôle. Amazon S3 calcule automatiquement la valeur du total de contrôle.
- Lorsque vous utilisez l'API REST, vous n'utilisez pas le paramètre `x-amz-sdk-checksum-algorithm`. Vous devez plutôt utiliser l'un des en-têtes spécifiques à l'algorithme (par exemple, `x-amz-checksum-crc32`).

Pour en savoir plus sur le chargement d'objets, consultez Chargement d'objets.

Pour appliquer l'une de ces valeurs de total de contrôle à des objets déjà chargés sur Amazon S3, vous pouvez copier l'objet. Lorsque vous copiez un objet, vous pouvez préciser si vous souhaitez utiliser l'algorithme de total de contrôle existant ou en utiliser un nouveau. Vous pouvez spécifier un algorithme de total de contrôle lors de l'utilisation de tout mécanisme pris en charge pour la copie d'objets, y compris les opérations par lots S3. Pour plus d'informations sur les opérations par lots S3, consultez Exécution des opérations par lot à grande échelle sur des objets Amazon S3.

> **Important**
>
> Si vous utilisez un chargement partitionné avec des totaux de contrôle supplémentaires, les numéros de partie partitionnés doivent utiliser des numéros de parties consécutifs. Lorsque vous utilisez des totaux de contrôle supplémentaires, si vous essayez de lancer une requête de chargement partitionné avec des numéros de parties non consécutifs, Amazon S3 génère une erreur `500 Internal Server Error` HTTP.

Après avoir chargé des objets, vous pouvez obtenir la valeur du total de contrôle et la comparer à une valeur de total de contrôle précalculée ou précédemment stockée, calculée à l'aide du même algorithme.

Utilisation de la console S3

Pour en savoir plus sur l'utilisation de la console et la définition des algorithmes de somme de contrôle à utiliser lors du chargement d'objets, consultez Chargement d'objets et Didacticiel : Vérification de l'intégrité des données dans Amazon S3 avec des sommes de contrôle supplémentaires.

Utilisation des AWS SDK

L'exemple suivant montre comment vous pouvez utiliser les AWS SDK pour télécharger un fichier volumineux avec un téléchargement en plusieurs parties, télécharger un fichier volumineux et valider un fichier de téléchargement en plusieurs parties, le tout en utilisant SHA-256 pour la validation des fichiers.

Java

Example Exemple : chargement, téléchargement et vérification d'un fichier volumineux avec SHA-256

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.ObjectAttributes;
import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.Tag;
import software.amazon.awssdk.services.s3.model.Tagging;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.ByteBuffer;
import java.security.MessageDigest;
```

```
    import java.security.NoSuchAlgorithmException;
    import java.util.ArrayList;
    import java.util.Base64;
    import java.util.List;

    public class LargeObjectValidation {
        private static String FILE_NAME = "sample.file";
        private static String BUCKET = "sample-bucket";
        //Optional, if you want a method of storing the full multipart object
 checksum in S3.
        private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
        //If you have existing full-object checksums that you need to validate
 against, you can do the full object validation on a sequential upload.
        private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/
mkgiAhXt5MFoVowVg+LE9aIQmI=";
        //Example Chunk Size - this must be greater than or equal to 5MB.
        private static int CHUNK_SIZE = 5 * 1024 * 1024;

        public static void main(String[] args) {
            S3Client s3Client = S3Client.builder()
                    .region(Region.US_EAST_1)
                    .credentialsProvider(new AwsCredentialsProvider() {
                        @Override
                        public AwsCredentials resolveCredentials() {
                            return new AwsCredentials() {
                                @Override
                                public String accessKeyId() {
                                    return Constants.ACCESS_KEY;
                                }

                                @Override
                                public String secretAccessKey() {
                                    return Constants.SECRET;
                                }
                            };
                        }
                    })
                    .build();
            uploadLargeFileBracketedByChecksum(s3Client);
            downloadLargeFileBracketedByChecksum(s3Client);
            validateExistingFileAgainstS3Checksum(s3Client);
        }

        public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
```

```
            System.out.println("Starting uploading file validation");
            File file = new File(FILE_NAME);
            try (InputStream in = new FileInputStream(file)) {
                MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
                CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
                        .bucket(BUCKET)
                        .key(FILE_NAME)
                        .checksumAlgorithm(ChecksumAlgorithm.SHA256)
                        .build();
                CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
                List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
                int partNumber = 1;
                byte[] buffer = new byte[CHUNK_SIZE];
                int read = in.read(buffer);
                while (read != -1) {
                    UploadPartRequest uploadPartRequest =
UploadPartRequest.builder()

.partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).ch
                    UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
                    CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256()
                    completedParts.add(part);
                    sha256.update(buffer, 0, read);
                    read = in.read(buffer);
                    partNumber++;
                }
                String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
                if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
                    //Because the SHA256 is uploaded after the part is uploaded; the
upload is bracketed and the full object can be fully validated.

s3Client.abortMultipartUpload(AbortMultipartUploadRequest.builder().bucket(BUCKET).key(FILE
                    throw new IOException("Byte mismatch between stored checksum and
upload, do not proceed with upload and cleanup");
                }
                CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
```

```
                CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(

CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUplo
                Tag checksumTag =
Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
                //Optionally, if you need the full object checksum stored with the
file; you could add it as a tag after completion.

s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).t
            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
            GetObjectAttributesResponse
                    objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
            System.out.println(objectAttributes.objectParts().parts());
            System.out.println(objectAttributes.checksum().checksumSHA256());
        }

        public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
            System.out.println("Starting downloading file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
            try (OutputStream out = new FileOutputStream(file)) {
                GetObjectAttributesResponse
                        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
                //Optionally if you need the full object checksum, you can grab a
tag you added on the upload
                List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).b
                String fullObjectChecksum = null;
                for (Tag objectTag : objectTags) {
                    if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                        fullObjectChecksum = objectTag.value();
                        break;
                    }
                }
                MessageDigest sha256FullObject =
MessageDigest.getInstance("SHA-256");
```

```
                MessageDigest sha256ChecksumOfChecksums =
 MessageDigest.getInstance("SHA-256");

                //If you retrieve the object in parts, and set the ChecksumMode to
 enabled, the SDK will automatically validate the part checksum
                for (int partNumber = 1; partNumber <=
 objectAttributes.objectParts().totalPartsCount(); partNumber++) {
                    MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                    ResponseInputStream<GetObjectResponse> response =
 s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(part
                    GetObjectResponse getObjectResponse = response.response();
                    byte[] buffer = new byte[CHUNK_SIZE];
                    int read = response.read(buffer);
                    while (read != -1) {
                        out.write(buffer, 0, read);
                        sha256FullObject.update(buffer, 0, read);
                        sha256Part.update(buffer, 0, read);
                        read = response.read(buffer);
                    }
                    byte[] sha256PartBytes = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(sha256PartBytes);
                    //Optionally, you can do an additional manual validation again
 the part checksum if needed in addition to the SDK check
                    String base64PartChecksum =
 Base64.getEncoder().encodeToString(sha256PartBytes);
                    String base64PartChecksumFromObjectAttributes =
 objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();
                    if (!
base64PartChecksum.equals(getObjectResponse.checksumSHA256()) || !
base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
                        throw new IOException("Part checksum didn't match for the
 part");
                    }
                    System.out.println(partNumber + " " + base64PartChecksum);
                }
                //Before finalizing, do the final checksum validation.
                String base64FullObject =
 Base64.getEncoder().encodeToString(sha256FullObject.digest());
                String base64ChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
                    throw new IOException("Failed checksum validation for full
 object");
```

```
                }
                System.out.println(fullObjectChecksum);
                String base64ChecksumOfChecksumFromAttributes =
 objectAttributes.checksum().checksumSHA256();
                if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
                    throw new IOException("Failed checksum validation for full
 object checksum of checksums");
                }
                System.out.println(base64ChecksumOfChecksumFromAttributes);
                out.flush();
            } catch (IOException | NoSuchAlgorithmException e) {
                //Cleanup bad file
                file.delete();
                e.printStackTrace();
            }
        }

        public static void validateExistingFileAgainstS3Checksum(S3Client s3Client)
 {
            System.out.println("Starting existing file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
            GetObjectAttributesResponse
                    objectAttributes =
 s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
 ObjectAttributes.CHECKSUM).build());
            try (InputStream in = new FileInputStream(file)) {
                MessageDigest sha256ChecksumOfChecksums =
 MessageDigest.getInstance("SHA-256");
                MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                byte[] buffer = new byte[CHUNK_SIZE];
                int currentPart = 0;
                int partBreak =
 objectAttributes.objectParts().parts().get(currentPart).size();
                int totalRead = 0;
                int read = in.read(buffer);
                while (read != -1) {
                    totalRead += read;
                    if (totalRead >= partBreak) {
                        int difference = totalRead - partBreak;
                        byte[] partChecksum;
                        if (totalRead != partBreak) {
                            sha256Part.update(buffer, 0, read - difference);
```

```
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                            sha256Part.update(buffer, read - difference,
 difference);
                        } else {
                            sha256Part.update(buffer, 0, read);
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                        }
                        String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                        if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSH
 {
                            throw new IOException("Part checksum didn't match S3");
                        }
                        currentPart++;
                        System.out.println(currentPart + " " + base64PartChecksum);
                        if (currentPart <
 objectAttributes.objectParts().totalPartsCount()) {
                            partBreak +=
 objectAttributes.objectParts().parts().get(currentPart - 1).size();
                        }
                    } else {
                        sha256Part.update(buffer, 0, read);
                    }
                    read = in.read(buffer);
                }
                if (currentPart != objectAttributes.objectParts().totalPartsCount())
 {
                    currentPart++;
                    byte[] partChecksum = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(partChecksum);
                    String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                    System.out.println(currentPart + " " + base64PartChecksum);
                }

                String base64CalculatedChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                System.out.println(base64CalculatedChecksumOfChecksums);
                System.out.println(objectAttributes.checksum().checksumSHA256());
```

```
                if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
 {
                    throw new IOException("Full object checksum of checksums don't
 match S3");
                }

            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
        }
    }
```

Utilisation de l'API REST

Vous pouvez envoyer des requêtes REST pour télécharger un objet avec une valeur de somme de contrôle afin de vérifier l'intégrité des PutObjectdonnées. Vous pouvez également récupérer la valeur de la somme de contrôle pour les objets à l'aide de GetObjectou HeadObject.

À l'aide du AWS CLI

Vous pouvez envoyer une demande PUT pour télécharger un objet d'une taille maximale de 5 Go en une seule opération. Pour plus d'informations, reportez-vous à la section `PutObject` dans la référence des commandes AWS CLI . Vous pouvez également utiliser `get-object` et `head-object` pour récupérer le total de contrôle d'un objet déjà chargé afin de vérifier l'intégrité des données.

Pour plus d'informations, consultez la FAQ de l'interface de ligne de commande Amazon S3 dans le guide de AWS Command Line Interface l'utilisateur.

Utilisation de Content-MD5 pour charger des objets

Une autre façon de vérifier l'intégrité de votre objet après le chargement est de fournir un récapitulatif MD5 de l'objet lorsque vous le chargez. Si vous calculez le récapitulatif MD5 de votre objet, vous pouvez fournir ce récapitulatif avec la commande PUT en utilisant l'en-tête `Content-MD5`.

Après avoir chargé l'objet, Amazon S3 calcule le récapitulatif MD5 de l'objet et le compare à la valeur que vous avez fournie. La requête n'aboutit que si les deux récapitulatifs correspondent.

La fourniture d'un récapitulatif MD5 n'est pas obligatoire, mais vous pouvez l'utiliser pour vérifier l'intégrité de l'objet dans le cadre du processus de chargement.

Utilisation de Content-MD5 et de ETag pour vérifier les objets chargés.

Le balise d'entité (ETag) d'un objet représente une version spécifique de cet objet. Gardez à l'esprit que ETag ne reflète que les changements apportés au contenu d'un objet et non à ses métadonnées. Si seules les métadonnées d'un objet changent, la balise ETag reste le même.

Selon l'objet, la balise ETag de l'objet peut être un récapitulatif MD5 des données de l'objet :

- Si un objet est créé par l'opération `PutObject`, `PostObject`, ou `CopyObject`, ou via la AWS Management Console et que cet objet est également en texte brut ou chiffré par un chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3), cet objet possède un ETag qui est un récapitulatif MD5 de ses données.
- Si un objet est créé par l'`CopyObject`opération `PutObject``PostObject`, ou par le biais de AWS Management Console, et que cet objet est chiffré par chiffrement côté serveur avec des clés fournies par le client (SSE-C) ou par chiffrement côté serveur avec des clés () (SSE-KMS AWS KMS), cet objet possède un ETag qui n'est pas un condensé MD5 de ses données d'objet. AWS Key Management Service
- Si un objet est créé par l'opération `Multipart Upload` ou `Part Copy`, la balise ETag de l'objet n'est pas un récapitulatif MD5, quelle que soit la méthode de chiffrement. Si un objet est plus grand que 16 Mo, le chargement AWS Management Console ou la copie de cet objet est un chargement partitionné et la balise ETag n'est pas un récapitulatif MD5.

Pour les objets où la balise ETag est le récapitulatif `Content-MD5` de l'objet, vous pouvez comparer la valeur ETag de l'objet avec un récapitulatif `Content-MD5` calculé ou précédemment stocké.

Utilisation des totaux de contrôle de fin

Lorsque vous chargez des objets sur Amazon S3, vous pouvez soit fournir une somme de contrôle précalculée pour l'objet, soit utiliser un AWS SDK pour créer automatiquement des sommes de contrôle de suivi en votre nom. Si vous décidez d'utiliser un total de contrôle de fin, Amazon S3 génère automatiquement le total de contrôle en utilisant l'algorithme que vous avez spécifié et l'utilise pour valider l'intégrité de l'objet pendant le chargement.

Pour créer une somme de contrôle finale lors de l'utilisation d'un AWS SDK, renseignez le `ChecksumAlgorithm` paramètre avec votre algorithme préféré. Le kit SDK utilise cet algorithme pour calculer le total de contrôle de votre objet (ou de ses parties) et l'ajoute automatiquement à la fin de votre requête de chargement. Ce comportement vous fait gagner du temps car Amazon S3 effectue la vérification et le chargement de vos données en une seule opération.

> **Important**
>
> Si vous utilisez S3 Object Lambda, toutes les requêtes adressées à S3 Object Lambda sont signées en utilisant `s3-object-lambda` au lieu de `s3`. Ce comportement affecte la signature des valeurs de total de contrôle de fin. Pour en savoir plus sur S3 Object Lambda, consultez Transformation d'objets avec S3 Object Lambda.

Utilisation de totaux de contrôle au niveau des parties pour les chargements partitionnés

Lorsque des objets sont chargés sur Amazon S3, ils peuvent être chargés en tant qu'objet unique ou par le biais du processus de chargement partitionné. Les objets dont la taille est supérieure à 16 Mo et qui sont chargés par le biais de la console sont automatiquement chargés à l'aide de chargements partitionnés. Pour en savoir plus sur le chargement partitionné, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

Lorsqu'un objet est chargé en tant que chargement partitionné, la balise ETag de l'objet n'est pas un récapitulatif MD5 de l'objet entier. Amazon S3 calcule le récapitulatif MD5 de chaque partie individuelle au fur et à mesure qu'elle est chargée. Les récapitulatifs MD5 sont utilisés pour déterminer la balise ETag de l'objet final. Amazon S3 concatène les octets des récapitulatifs MD5, puis calcule le récapitulatif MD5 de ces valeurs concaténées. L'étape finale de la création de la balise ETag est celle où Amazon S3 ajoute à la fin un tiret avec le nombre total de parties.

Par exemple, prenons un objet chargé avec un chargement partitionné qui possède un ETag de `C9A5A6878D97B48CC965C1E41859F034-14`. Dans ce cas, `C9A5A6878D97B48CC965C1E41859F034` représente le récapitulatif MD5 de tous les récapitulatifs concaténés ensemble. `-14` indique que 14 parties sont associées au chargement partitionné de cet objet.

Si vous avez activé des valeurs de contrôle supplémentaires pour votre objet en plusieurs parties, Amazon S3 calcule le total de contrôle pour chaque partie individuelle en utilisant l'algorithme de total de contrôle spécifié. Le total de contrôle pour l'objet terminé est calculé de la même manière que Amazon S3 calcule le récapitulatif MD5 pour le chargement partitionné. Vous pouvez utiliser ce total de contrôle pour vérifier l'intégrité de l'objet.

Pour récupérer des informations sur l'objet, notamment le nombre de parties composant l'objet entier, vous pouvez utiliser cette GetObjectAttributesopération. Avec des totaux de contrôle

supplémentaires, vous pouvez également récupérer des informations pour chaque partie individuelle qui incluent la valeur du total de contrôle de chaque partie.

Pour les téléchargements terminés, vous pouvez obtenir la somme de contrôle d'une partie individuelle en utilisant les HeadObjectopérations GetObjector et en spécifiant un numéro de pièce ou une plage d'octets alignée sur une seule partie. Si vous souhaitez récupérer les valeurs de contrôle pour les différentes parties des téléchargements partitionnés toujours en cours, vous pouvez utiliser. ListParts

En raison de la façon dont Amazon S3 calcule le total de contrôle pour les objets à plusieurs parties, la valeur du total de contrôle de l'objet peut changer si vous le copiez. Si vous utilisez un SDK ou l'API REST et que vous appelez CopyObject, Amazon S3 copie n'importe quel objet dans les limites de taille de l'opération d'`CopyObject`API. Amazon S3 effectue cette copie en une seule action, que l'objet ait été chargé en une seule requête ou dans le cadre d'un chargement partitionné. Avec une commande de copie, le total de contrôle de l'objet est un total de contrôle direct de l'objet complet. Si l'objet a été initialement chargé à l'aide d'un chargement partitionné, la valeur du total de contrôle change même si les données ne changent pas.

> **Note**
>
> Les objets dont la taille dépasse les limites de l'opération API `CopyObject` doivent utiliser des commandes de copie en plusieurs parties.

> **Important**
>
> Lorsque vous effectuez certaines opérations à l'aide du AWS Management Console, Amazon S3 utilise un téléchargement partitionné si la taille de l'objet est supérieure à 16 Mo. Dans ce cas, le total de contrôle n'est pas un total de contrôle direct de l'objet complet, mais plutôt un calcul basé sur les valeurs de total de contrôle de chaque partie individuelle.
> Prenons l'exemple d'un objet de 100 Mo que vous avez chargé directement en une seule partie à l'aide de l'API REST. Dans ce cas, le total de contrôle est un total de contrôle de l'objet entier. Si vous utilisez ensuite la console pour renommer cet objet, le copier, changer sa classe de stockage ou modifier les métadonnées, Amazon S3 utilise la fonctionnalité de chargement partitionné pour mettre à jour l'objet. Par conséquent, Amazon S3 crée une nouvelle valeur de total de contrôle pour l'objet qui est calculée sur la base des valeurs de total de contrôle des parties individuelles.

La liste précédente des opérations de console n'est pas une liste complète de toutes les actions possibles que vous pouvez entreprendre pour AWS Management Console qu'Amazon S3 mette à jour l'objet à l'aide de la fonctionnalité de téléchargement partitionné. Gardez à l'esprit que lorsque vous utilisez la console pour agir sur des objets d'une taille supérieure à 16 Mo, la valeur du total de contrôle peut ne pas correspondre au total de contrôle de l'objet entier.

Suppression d'objets Amazon S3

Vous pouvez supprimer un ou plusieurs objets directement depuis Amazon S3 à l'aide de la console Amazon S3, AWS des SDK AWS Command Line Interface (AWS CLI) ou de l'API REST. Etant donné que tous les objets dans le compartiment S3 entraînent des coûts de stockage, vous devez supprimer ceux dont vous n'avez plus besoin. Par exemple, si vous collectez des fichiers journaux, vous pouvez les supprimer lorsque vous n'en avez plus besoin. Vous pouvez configurer une règle de cycle de vie pour supprimer automatiquement les objets tels que les fichiers journaux. Pour plus d'informations, consultez the section called “Définition d'une configuration de cycle de vie”.

Pour en savoir plus sur les fonctions et la tarification d'Amazon S3, veuillez consulter Tarification Amazon S3.

Lors de la suppression d'un objet, vous disposez des options d'API suivantes :

- Supprimer un seul objet : Amazon S3 fournit l'opération d'API `DELETE` (`DeleteObject`) que vous pouvez utiliser pour supprimer un objet en une seule demande HTTP.
- Supprimer plusieurs objets : Amazon S3 fournit l'opération d'API de suppression de plusieurs objets (`DeleteObjects`) que vous pouvez utiliser pour supprimer jusqu'à 1,000 objets en une seule demande HTTP.

Lorsque vous supprimez des objets d'un compartiment qui n'est pas activé pour le contrôle de version, vous ne fournissez que le nom de la clé d'objet. Toutefois, lorsque vous supprimez des objets d'un compartiment activé pour le contrôle de version, vous pouvez éventuellement fournir l'ID de version de l'objet pour supprimer une version spécifique de l'objet.

Suppression par programme d'objets d'un compartiment activé pour le contrôle de version

Si le contrôle de version est activé pour votre compartiment, plusieurs versions du même objet peuvent exister dans le compartiment. Si vous utilisez des compartiments pour lesquels la gestion des versions a été activée, les opérations d'API DELETE permettent les options suivantes :

- Spécifier une demande de suppression sans la gestion des versions : spécifiez uniquement la clé de l'objet, et non l'ID de la version. Dans ce cas, Amazon S3 crée un marqueur de suppression et renvoie son ID de version dans la réponse. Votre objet n'apparaît plus dans le compartiment. Pour plus d'informations sur le contrôle de version d'objets et le concept des marqueurs de suppression, consultez Utilisation de la gestion des versions dans les compartiments S3.
- Spécifier une demande de suppression avec la gestion des versions : spécifiez à la fois la clé et l'ID de version. Dans ce cas, les deux résultats suivants sont possibles :
  - Si l'ID de version est lié à une version d'objet spécifique, Amazon S3 supprime la version spécifique de l'objet.
  - Si l'ID de version est lié au marqueur de suppression de cet objet, Amazon S3 supprime le marqueur de suppression. L'objet réapparaît alors dans votre compartiment.

Suppression des objets d'un compartiment avec authentification MFA activée

Lorsque vous supprimez des objets d'un compartiment prenant en charge l'authentification multifacteur (MFA), notez les points suivants :

- Si vous fournissez un jeton d'authentification MFA non valide, la demande échouera.
- Si vous avez un compartiment activé pour l'authentification MFA et que vous faites une demande de suppression avec la gestion de version (vous fournissez une clé d'objet et un ID de version), la demande échoue si vous ne fournissez pas un jeton d'authentification MFA valide. En outre, avec l'opération d'API de suppression de plusieurs objets dans un compartiment activé pour l'authentification MFA, si l'un des objets supprimés est une demande de suppression avec la gestion des versions (à savoir, vous indiquez la clé de l'objet et l'ID de version), la demande échoue dans son intégralité si vous ne fournissez pas un jeton d'authentification MFA.

Toutefois, dans les cas suivants, la demande réussit :

- Si vous avez un compartiment pour lequel l'authentification MFA est activée et que vous faites une demande de suppression sans la gestion des versions (vous ne supprimez pas un objet avec la gestion des versions), et que vous ne fournissez pas de jeton d'authentification MFA, la suppression aboutit.
- Si vous avez une demande de suppression de plusieurs objets spécifiant des objets sans la gestion des versions à supprimer d'un compartiment pour lequel l'authentification MFA est activée, et que vous ne fournissez pas de jeton d'authentification MFA, la suppression aboutit.

Pour en savoir plus sur la suppression MFA, veuillez consulter Configuration de la fonction Supprimer MFA.

Rubriques

- Suppression d'un seul objet
- Suppression de plusieurs objets

Suppression d'un seul objet

Vous pouvez utiliser la console Amazon S3 ou l'API DELETE pour supprimer un seul objet existant d'un compartiment S3. Pour plus d'informations sur la suppression d'objets dans Amazon S3, consultez Suppression d'objets Amazon S3.

Etant donné que tous les objets dans le compartiment S3 entraînent des coûts de stockage, vous devez supprimer ceux dont vous n'avez plus besoin. Par exemple, si vous collectez des fichiers journaux, vous pouvez les supprimer lorsque vous n'en avez plus besoin. Vous pouvez configurer une règle de cycle de vie pour supprimer automatiquement les objets tels que les fichiers journaux. Pour plus d’informations, consultez the section called “Définition d'une configuration de cycle de vie”.

Pour en savoir plus sur les fonctions et la tarification d'Amazon S3, veuillez consulter Tarification Amazon S3.

Utiliser la console S3.

Procédez comme suit pour utiliser la console Amazon S3 pour supprimer un seul objet d'un compartiment.

> **Warning**
>
> Lorsque vous supprimez définitivement un objet ou une version d'objet spécifiée dans la console Amazon S3, la suppression ne peut pas être annulée.

Pour supprimer un objet dont le contrôle de version est activé ou suspendu

> **Note**
>
> Si l'ID de version d'un objet dans un compartiment suspendu aux versions est marqué comme telNULL, S3 supprime définitivement l'objet car aucune version précédente n'existe. Toutefois, si un ID de version valide est répertorié pour l'objet dans un compartiment suspendu aux versions, S3 crée un marqueur de suppression pour l'objet supprimé, tout en conservant les versions précédentes de l'objet.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment à partir duquel vous souhaitez supprimer un objet.
3. Sélectionnez l'objet, puis choisissez Supprimer.
4. Pour confirmer la suppression de la liste d'objets sous Objets spécifiés dans la section Supprimer des objets ? zone de texte, entrez**`delete`**.

Pour supprimer définitivement une version d'objet spécifique dans un compartiment activé pour la gestion des versions

> **Warning**
>
> Lorsque vous supprimez définitivement une version d'objet spécifique dans Amazon S3, la suppression ne peut pas être annulée.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans la liste Nom du compartiment, choisissez le nom du compartiment à partir duquel vous souhaitez supprimer un objet.
3. Sélectionnez l'objets que vous voulez supprimer.
4. Choisissez le bouton Afficher les versions.
5. Sélectionnez la version de l'objet, puis choisissez Supprimer.
6. Pour confirmer la suppression définitive des versions d'objets spécifiques répertoriées sous Objets spécifiés, dans la section Supprimer des objets ? zone de texte, saisissez Supprimer définitivement. Amazon S3 supprime définitivement la version de l'objet spécifique.

Pour supprimer définitivement un objet dans un compartiment Amazon S3 pour lequel le contrôle de version n'est pas activé

> **Warning**
>
> Lorsque vous supprimez définitivement un objet dans Amazon S3, la suppression ne peut pas être annulée. De plus, pour tous les compartiments dont le versionnement n'est pas activé, les suppressions sont permanentes.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment à partir duquel vous souhaitez supprimer un objet.
3. Sélectionnez l'objet, puis choisissez Supprimer.
4. Pour confirmer la suppression définitive de l'objet répertorié sous Objets spécifiés, dans la section Supprimer des objets ? zone de texte, entrez Supprimer définitivement.

> **Note**
>
> Si vous rencontrez des problèmes lors de la suppression de votre objet, consultezJe souhaite supprimer définitivement les objets avec la gestion des versions.

Utilisation des AWS SDK

Les exemples suivants montrent comment utiliser les AWS SDK pour supprimer un objet d'un bucket. Pour plus d'informations, consultez Objet DELETE dans la Référence d'API Amazon Simple Storage Service.

Si le compartiment est activé pour le contrôle de version S3, vous disposez des options suivantes :

- Supprimez une version spécifique d'objet en spécifiant un ID de version.
- Supprimez un objet sans spécifier un ID de version, auquel cas Amazon S3 ajoute un marqueur de suppression à l'objet.

Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3.

Java

Example Exemple 1 : Suppression d'un objet (compartiment désactivé pour la gestion des versions)

L'exemple suivant suppose que le compartiment n'est pas activé pour le contrôle de version et qu'il n'a pas des ID de version. Dans la demande de suppression, vous ne spécifiez que la clé d'objet, et pas d'ID de version.

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

import java.io.IOException;

public class DeleteObjectNonVersionedBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
```

```
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            s3Client.deleteObject(new DeleteObjectRequest(bucketName, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Exemple 2 : Suppression d'un objet (compartiment activé pour la gestion des versions)

L'exemple suivant supprime un objet à partir d'un compartiment activé pour la gestion des versions. L'exemple supprime une version d'objet spécifique en spécifiant le nom de la clé d'objet et l'ID de version.

Cet exemple effectue les opérations suivantes :

1. Ajoute un exemple d'objet au compartiment. Amazon S3 renvoie l'ID de version de l'objet nouvellement ajouté. L'exemple utilise l'ID de version dans la demande de suppression.
2. Supprime la version d'objet spécifique en spécifiant le nom de la clé d'objet et l'ID de version. S'il n'existe pas d'autres versions de cet objet, Amazon S3 supprime la totalité de l'objet. Sinon, Amazon S3 supprime uniquement la version spécifiée.

> **Note**
>
> Vous pouvez également obtenir les ID de version d'un objet en envoyant une demande `ListVersions`.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteVersionRequest;
import com.amazonaws.services.s3.model.PutObjectResult;

import java.io.IOException;

public class DeleteObjectVersionEnabledBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
 s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED))
 {
                System.out.printf("Bucket %s is not versioning-enabled.",
 bucketName);
            } else {
                // Add an object.
                PutObjectResult putResult = s3Client.putObject(bucketName, keyName,
                        "Sample content for deletion example.");
                System.out.printf("Object %s added to bucket %s\n", keyName,
 bucketName);

                // Delete the version of the object that we just created.
                System.out.println("Deleting versioned object " + keyName);
```

```
                s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
 putResult.getVersionId()));
                System.out.printf("Object %s, version %s deleted\n", keyName,
 putResult.getVersionId());
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Les exemples suivants montrent comment supprimer un objet à partir d'un compartiment activé pour la gestion des versions et d'un compartiment non activé pour la gestion des versions. Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3.

Example Suppression d'un objet d'un compartiment désactivé pour la gestion des versions

L'exemple C# suivant supprime un objet à partir d'un compartiment désactivé pour la gestion des versions. Comme l'exemple suppose que les objets n'ont pas d'ID de version, ne spécifiez pas d'ID de version. Spécifiez uniquement la clé d'objet.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
```

```
    class DeleteObjectNonVersionedBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            DeleteObjectNonVersionedBucketAsync().Wait();
        }
        private static async Task DeleteObjectNonVersionedBucketAsync()
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(deleteObjectRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
        }
    }
}
```

Example Suppression d'un objet d'un compartiment activé pour la gestion des versions

L'exemple C# suivant supprime un objet d'un compartiment activé pour la gestion des versions. L'exemple supprime une version de l'objet en spécifiant le nom de la clé d'objet et l'ID de version.

Le code exécute les tâches suivantes :

1. Active le contrôle de version S3 sur un compartiment que vous spécifiez (si le contrôle de version S3 est déjà activé, cette opération est sans effet).
2. Ajoute un exemple d'objet au compartiment. En réponse, Amazon S3 renvoie l'ID de version de l'objet nouvellement ajouté. L'exemple utilise l'ID de version dans la demande de suppression.
3. Supprime l'exemple d'objet en spécifiant le nom de la clé d'objet et l'ID de version.

> **Note**
>
> Vous pouvez également obtenir l'ID de version d'un objet en envoyant une demande `ListVersions`.
>
> ```
> var listResponse = client.ListVersions(new ListVersionsRequest { BucketName
>  = bucketName, Prefix = keyName });
> ```

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectVersion
    {
        private const string bucketName = "*** versioning-enabled bucket name ***";
        private const string keyName = "*** Object Key Name ***";
        // Specify your bucket region (an example region is shown).
```

```
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateAndDeleteObjectVersionAsync().Wait();
        }

        private static async Task CreateAndDeleteObjectVersionAsync()
        {
            try
            {
                // Add a sample object.
                string versionID = await PutAnObject(keyName);

                // Delete the object by specifying an object key and a version ID.
                DeleteObjectRequest request = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    VersionId = versionID
                };
                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
        }

        static async Task<string> PutAnObject(string objectKey)
        {
            PutObjectRequest request = new PutObjectRequest
            {
                BucketName = bucketName,
```

```
                Key = objectKey,
                ContentBody = "This is the content body!"
            };
            PutObjectResponse response = await client.PutObjectAsync(request);
            return response.VersionId;
        }
    }
}
```

PHP

Cet exemple montre comment utiliser les classes de la version 3 de AWS SDK for PHP pour supprimer un objet d'un bucket non versionné. Pour en savoir plus sur la suppression d'un objet à partir d'un compartiment activé pour le contrôle de version, consultez Utilisation de l'API REST.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple de PHP suivant supprime un objet à partir d'un compartiment. Comme cet exemple illustre la façon de supprimer des objets des compartiments désactivés pour le contrôle de version, il fournit uniquement le nom du compartiment et la clé d'objet (pas un ID de version) dans la demande de suppression.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// 1. Delete the object from the bucket.
try
{
    echo 'Attempting to delete ' . $keyname . '...' . PHP_EOL;

    $result = $s3->deleteObject([
```

```
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    if ($result['DeleteMarker'])
    {
        echo $keyname . ' was deleted or does not exist.' . PHP_EOL;
    } else {
        exit('Error: ' . $keyname . ' was not deleted.' . PHP_EOL);
    }
}
catch (S3Exception $e) {
    exit('Error: ' . $e->getAwsErrorMessage() . PHP_EOL);
}

// 2. Check to see if the object was deleted.
try
{
    echo 'Checking to see if ' . $keyname . ' still exists...' . PHP_EOL;

    $result = $s3->getObject([
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    echo 'Error: ' . $keyname . ' still exists.';
}
catch (S3Exception $e) {
    exit($e->getAwsErrorMessage());
}
```

Javascript

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js" // Helper function that creates Amazon
 S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };

export const run = async () => {
  try {
    const data = await s3Client.send(new DeleteObjectCommand(bucketParams));
    console.log("Success. Object deleted.", data);
```

```
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

À l'aide du AWS CLI

Pour supprimer un objet par demande, utilisez l'API DELETE. Pour plus d'informations, consultez Objet DELETE. Pour plus d'informations sur l'utilisation de la CLI pour supprimer un objet, consultez delete-object.

Utilisation de l'API REST

Vous pouvez utiliser les AWS SDK pour supprimer un objet. Toutefois, si l'application l'exige, vous pouvez envoyer directement des demandes REST. Pour plus d'informations, consultez Objet DELETE dans la Référence d'API Amazon Simple Storage Service.

Suppression de plusieurs objets

Etant donné que tous les objets dans le compartiment S3 entraînent des coûts de stockage, vous devez supprimer ceux dont vous n'avez plus besoin. Par exemple, si vous collectez des fichiers journaux, vous pouvez les supprimer lorsque vous n'en avez plus besoin. Vous pouvez configurer une règle de cycle de vie pour supprimer automatiquement les objets tels que les fichiers journaux. Pour plus d'informations, consultez the section called "Définition d'une configuration de cycle de vie".

Pour en savoir plus sur les fonctions et la tarification d'Amazon S3, veuillez consulter Tarification Amazon S3.

Vous pouvez utiliser la console Amazon S3, AWS les kits SDK ou l'API REST pour supprimer plusieurs objets simultanément d'un compartiment S3.

Utilisation de la console S3

Procédez comme suit pour utiliser la console Amazon S3 afin de supprimer plusieurs objets d'un compartiment.

Warning

- La suppression d'un objet spécifié ne peut pas être annulée.
- Cette action supprime tous les objets spécifiés. Lorsque vous supprimez des dossiers, attendez la fin de l'action de suppression pour ajouter de nouveaux objets au dossier. Dans le cas contraire, de nouveaux objets pourraient également être supprimés.
- Lorsque vous supprimez des objets dans un compartiment sans que le versionnement soit activé, Amazon S3 supprime définitivement les objets.
- Lorsque vous supprimez des objets dans un compartiment dont le versionnement des compartiments est activé ou suspendu, Amazon S3 crée des marqueurs de suppression. Pour en savoir plus, consultez Utilisation des marqueurs de suppression.

Pour supprimer des objets dont le contrôle de version est activé ou suspendu

Note

Si les ID de version de l'objet dans un compartiment suspendu aux versions sont marqués comme telsNULL, S3 supprime définitivement les objets car aucune version précédente n'existe. Toutefois, si un ID de version valide est répertorié pour les objets d'un compartiment dont les versions sont suspendues, S3 crée des marqueurs de suppression pour les objets supprimés, tout en conservant les versions précédentes des objets.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment dont vous souhaitez supprimer les objets.
3. Sélectionnez les objets, puis choisissez Supprimer.
4. Pour confirmer la suppression de la liste d'objets sous Objets spécifiés dans la section Supprimer des objets ? zone de texte, entrez**`delete`**.

Pour supprimer définitivement des versions d'objets spécifiques dans un compartiment activé pour la gestion des versions

> **Warning**
>
> Lorsque vous supprimez définitivement des versions d'objets spécifiques dans Amazon S3, la suppression ne peut pas être annulée.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment dont vous souhaitez supprimer les objets.
3. Sélectionnez les objets que vous voulez supprimer.
4. Choisissez le bouton Afficher les versions.
5. Sélectionnez les versions de l'objet, puis choisissez Supprimer.
6. Pour confirmer la suppression définitive des versions d'objets spécifiques répertoriées sous Objets spécifiés, dans la section Supprimer des objets ? zone de texte, saisissez Supprimer définitivement. Amazon S3 supprime définitivement les versions d'objets spécifiques.

Pour supprimer définitivement les objets d'un compartiment Amazon S3 pour lesquels la gestion des versions n'est pas activée

> **Warning**
>
> Lorsque vous supprimez définitivement un objet dans Amazon S3, la suppression ne peut pas être annulée. De plus, pour tous les compartiments dont le versionnement n'est pas activé, les suppressions sont permanentes.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment dont vous souhaitez supprimer les objets.
3. Sélectionnez les objets, puis choisissez Supprimer.

4. Pour confirmer la suppression définitive des objets répertoriés sous Objets spécifiés, dans la section Supprimer des objets ? zone de texte, entrez Supprimer définitivement.

> **Note**
>
> Si vous rencontrez des problèmes lors de la suppression de vos objets, consultezJe souhaite supprimer définitivement les objets avec la gestion des versions.

Utilisation des AWS SDK

Pour des exemples de suppression de plusieurs objets à l'aide AWS des SDK, consultezUtilisation `DeleteObjects` avec un AWS SDK ou une CLI.

Pour des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Utilisation de l'API REST

Vous pouvez utiliser les AWS SDK pour supprimer plusieurs objets à l'aide de l'API Multi-Object Delete. Toutefois, si l'application l'exige, vous pouvez envoyer directement des demandes REST.

Pour plus d'informations, consultez Suppression de plusieurs objets dans la Référence d'API Amazon Simple Storage Service.

Organisation, liste et utilisation de vos objets

Dans Amazon S3, vous pouvez utiliser des préfixes pour organiser votre stockage. Un préfixe est un regroupement logique des objets dans un compartiment. La valeur de préfixe est semblable à un nom de répertoire qui vous permet de stocker des données similaires sous le même répertoire d'un compartiment. Lorsque vous chargez des objets par programme, vous pouvez utiliser des préfixes pour organiser vos données.

Dans la console Amazon S3, les préfixes sont appelés dossiers. Vous pouvez afficher tous vos objets et dossiers dans la console S3 en accédant à un compartiment. Vous pouvez également afficher des informations sur chaque objet, y compris les propriétés de l'objet.

Pour en savoir plus sur l'élaboration de liste et l'organisation de vos données dans Amazon S3, consultez les rubriques suivantes.

Rubriques

Organisation des objets à l'aide de préfixes

Vous pouvez utiliser des préfixes pour organiser les données que vous stockez dans des compartiments Amazon S3. Un préfixe est une chaîne de caractères au début du nom de la clé d'objet. Il n'a pas de limite de longueur, mais ne peut pas dépasser la longueur du nom de la clé d'objet (1 024 octets). Vous pouvez voir les préfixes comme un moyen d'organiser vos données de la même manière que les répertoires. Toutefois, les préfixes ne sont pas des répertoires.

La recherche par préfixe limite les résultats aux seules clés qui commencent par le préfixe spécifié. Le délimiteur permet à la liste de regrouper toutes les clés qui partagent un préfixe commun dans une seule liste de résultats récapitulative.

L'objectif des paramètres de préfixe et de délimiteur est de vous aider à organiser et à parcourir hiérarchiquement les clés. Pour ce faire, choisissez tout d'abord un délimiteur pour le compartiment, comme une barre oblique (/), qui n'est présent dans aucun des noms de clés anticipés. Vous pouvez utiliser un autre caractère comme délimiteur. La barre oblique (/) n'est pas unique, mais elle est un délimiteur de préfixe très courant. Ensuite, créez les noms de clés en concaténant tous les niveaux de contenu de la hiérarchie, en séparant chaque niveau avec le délimiteur.

Par exemple, si vous stockez des informations sur des villes, vous devez naturellement les organiser par continent, puis par pays, puis par province ou état. Étant donné que ces noms ne contiennent généralement pas de ponctuation, vous devez utiliser une barre oblique (/) comme délimiteur. Les exemples suivants utilisent une barre oblique (/) comme délimiteur.

- Europe/France/Nouvelle-Aquitaine/Bordeaux
- Amérique du Nord/Canada/Québec/Montréal
- Amérique du Nord/États-Unis/Washington/Bellevue
- Amérique du Nord/États-Unis/Washington/Seattle

Si vous stockez des données pour chaque ville du monde de cette façon, il apparaît curieux de gérer un espace de noms de clé horizontal. Grâce au `Prefix` et au `Delimiter` avec l'opération de la liste, vous pouvez utiliser la hiérarchie que vous avez créée pour lister les données. Par exemple, pour lister tous les États des États-Unis, configurez `Delimiter='/'` et `Prefix='North America/USA/'`. Pour lister toutes les provinces du Canada pour lesquelles vous avez des données, configurez `Delimiter='/'` et `Prefix='North America/Canada/'`.

Pour plus d'informations sur les délimiteurs, les préfixes et les dossiers imbriqués, consultez Difference between prefixes and nested folders (Différence entre les préfixes et les dossiers imbriqués).

Liste d'objets à l'aide de préfixes et de délimiteurs

Si vous demandez une liste avec un délimiteur, vous pouvez parcourir la hiérarchie sur un seul niveau en passant et en résumant les (éventuels millions de) clés qui se trouvent dans les niveaux suivants. Par exemple, supposons que vous avez un compartiment (*DOC-EXAMPLE-BUCKET*) avec les clés suivantes :

```
sample.jpg

photos/2006/January/sample.jpg

photos/2006/February/sample2.jpg

photos/2006/February/sample3.jpg

photos/2006/February/sample4.jpg
```

L'exemple de compartiment ne dispose que de l'objet `sample.jpg` à la racine. Pour lister uniquement les objets au niveau racine du compartiment, vous envoyez une demande GET sur le compartiment avec la barre oblique (`/`) comme délimiteur. En réponse, Simple Storage Service (Amazon S3) renvoie la clé de l'objet `sample.jpg` car elle ne contient pas le délimiteur `/`. Toutes les autres clés contiennent le délimiteur. Simple Storage Service (Amazon S3) regroupe ces clés et renvoie un seul élément `CommonPrefixes` avec la valeur de préfixe `photos/`, qui est une sous-chaîne du début de ces clés jusqu'à la première occurrence du délimiteur spécifié.

Example

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

```
  <Name>DOC-EXAMPLE-BUCKET</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-07-24T19:39:30.000Z</LastModified>
    <ETag>&quot;d1a7fb5eab1c16cb4f7cf341cf188c3d&quot;</ETag>
    <Size>6</Size>
    <Owner>
      <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>displayname</DisplayName>
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Pour plus d'informations sur la liste des clés d'objet par programme, veuillez consulter Liste des clés d'objet par programme.

Liste des clés d'objet par programme

Dans Amazon S3, les clés peuvent être répertoriées par préfixe. Vous pouvez choisir un préfixe commun pour les noms des clés associées et marquer ces clés avec un caractère spécial délimitant la hiérarchie. Vous pouvez ensuite utiliser l'opération de liste pour sélectionner et parcourir les clés de manière hiérarchique. Cette méthode ressemble au stockage des fichiers dans les répertoires d'un système de fichiers.

Amazon S3 expose une opération de liste qui vous permet d'énumérer les clés contenues dans un compartiment. Les clés sont sélectionnées par compartiment et préfixe. Par exemple, prenez un compartiment appelé « `dictionary` » qui contient une clé pour chaque mot anglais. Vous devez faire un appel pour lister toutes les clés dans le compartiment qui commencent par la lettre « q ». Les résultats de la liste sont toujours renvoyés dans un ordre binaire UTF-8.

Les opérations de la liste SOAP et REST renvoient un document XML qui contient les noms des clés correspondantes et les informations sur l'objet identifié par chaque clé.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Au lieu d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Les groupes de clés qui partagent un préfixe se terminant par un délimiteur spécial peuvent être regroupés par ce préfixe commun pour les besoins de la liste. Cela permet aux applications d'organiser et de parcourir hiérarchiquement les clés, comme vous organiseriez des fichiers dans les répertoires d'un système de fichiers.

Par exemple, pour que le compartiment dictionnaire contienne plus que des mots anglais, vous devez former des clés en ajoutant un préfixe à chaque mot avec sa langue et un délimiteur, comme « `French/logical` ». Grâce à cette dénomination et à la fonction de liste hiérarchique, vous pouvez récupérer une liste de mots français. Vous pouvez également parcourir la liste de niveau supérieur des langues disponibles sans avoir à itérer sur toutes les clés d'intervention de manière lexicographique. Pour en savoir plus sur cet aspect de la liste, veuillez consulter Organisation des objets à l'aide de préfixes.

API REST

Si l'application l'exige, vous pouvez envoyer les demandes REST directement. Vous pouvez envoyer une demande GET pour renvoyer tout ou partie des objets d'un compartiment. Vous pouvez aussi utiliser des critères de sélection pour renvoyer un sous-ensemble d'objets d'un compartiment. Pour plus d'informations, consultez GET Bucket (List Objects) version 2 dans la Référence d'API Amazon Simple Storage Service..

Efficacité de l'implémentation d'une liste

Les performances de la liste ne sont pas significativement affectées par le nombre total de clés dans votre compartiment. Elles ne sont pas non plus affectées par la présence ou l'absence des arguments `prefix`, `marker`, `maxkeys`, ou `delimiter`.

Itération sur des résultats de plusieurs pages

Etant donné que les compartiments peuvent contenir un nombre presque illimité de clés, les résultats complets d'une requête de liste peuvent être très importants. Pour gérer d'importants ensembles de résultats, l'API Amazon S3 prend en charge la pagination afin de les scinder en plusieurs réponses.

Chaque réponse de clés de liste renvoie une page de 1 000 clés maximum avec un indicateur spécifiant si la réponse est tronquée. Vous envoyez une série de demandes de clés de liste jusqu'à ce que vous ayez reçu toutes les clés. AWS Les bibliothèques wrapper du SDK fournissent la même pagination.

Exemples

Les exemples de code suivants montrent comment utiliser`ListObjects`.

CLI

AWS CLI

L'exemple suivant utilise la `list-objects` commande pour afficher les noms de tous les objets du compartiment spécifié :

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

L'exemple utilise l'`--query`argument pour filtrer la sortie `list-objects` jusqu'à la valeur clé et à la taille de chaque objet.

Pour plus d'informations sur les objets, consultez la section Travailler avec des objets Amazon S3 dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous ListObjectsà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande récupère les informations relatives à tous les éléments du bucket « test-files ».

```
Get-S3Object -BucketName test-files
```

Exemple 2 : Cette commande récupère les informations relatives à l'élément « sample.txt » depuis le bucket « test-files ».

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Exemple 3 : Cette commande récupère les informations relatives à tous les éléments portant le préfixe « sample » à partir du bucket « test-files ».

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Pour plus de détails sur l'API, consultez la section ListObjectsRéférence des AWS Tools for PowerShell applets de commande.

Organisation des objets dans la console Amazon S3 à l'aide de dossiers

Dans Amazon S3, les compartiments et les objets sont les ressources principales, et les objets sont stockés dans des compartiments. Amazon S3 possède une structure horizontale et non une hiérarchie comme dans un système de fichiers. Toutefois, par souci de simplification organisationnelle, la console Amazon S3 supporte le concept de dossier comme moyen de regrouper des objets. Pour ce faire, la console utilise un préfixe de nom partagé pour les objets groupés. En d'autres termes, les objets de groupe ont des noms qui commencent par une chaîne commune. Cette chaîne commune, ou préfixe partagé, est le nom du dossier. Les noms d'objet sont également appelés noms de clé.

Par exemple, vous pouvez créer un dossier dans la console appelé `photos` et y stocker un objet nommé `myphoto.jpg`. L'objet est ensuite stocké avec le nom de clé `photos/myphoto.jpg`, dans lequel le préfixe est `photos/`.

Voici deux exemples supplémentaires :

- Si vous possédez trois objets dans le compartiment : `logs/date1.txt`, `logs/date2.txt` et `logs/date3.txt`, la console affiche un dossier appelé `logs`. Si vous ouvrez le dossier dans la console, vous voyez les trois objets : `date1.txt`, `date2.txt` et `date3.txt`.
- Si vous avez un objet nommé `photos/2017/example.jpg`, la console affichera un dossier appelé `photos` qui contient le dossier `2017`. Le dossier `2017` contiendra l'objet `example.jpg`.

Vous pouvez avoir des dossiers dans d'autres dossiers, mais pas de compartiments dans d'autres compartiments. Vous pouvez charger et copier des objets directement dans un dossier. Des dossiers peuvent être créés, supprimés et rendus publics, mais ils ne peuvent pas être renommés. Des objets peuvent être copiés d'un dossier vers un autre.

> **Important**
>
> Lorsque vous créez un dossier dans Amazon S3, ce dernier crée un objet de 0 octet dont la clé est définie par le nom de dossier que vous avez fourni. Par exemple, si vous créez un dossier nommé `photos` dans votre compartiment, la console Amazon S3 crée un objet de 0 octet avec la clé `photos/`. La console crée cet objet pour prendre en charge les dossiers. La console Amazon S3 traite tous les objets dont le nom de clé possède une barre oblique (/) à la fin comme un dossier (par exemple, `examplekeyname/`). Vous ne pouvez pas charger un objet qui possède un nom de clé avec un caractère / à la fin à l'aide de la console Amazon S3. Cependant, vous pouvez télécharger des objets nommés avec un suivi / avec l'API Amazon S3 à l'aide de l'API AWS Command Line Interface (AWS CLI), des AWS SDK ou de l'API REST.
>
> Un objet dont le nom se termine par un caractère / apparaît en tant que dossier dans la console Amazon S3. La console Amazon S3 n'affiche pas le contenu et les métadonnées pour un tel objet. Lorsque vous utilisez la console pour copier un objet dont le nom se termine par /, un nouveau dossier est créé dans l'emplacement de destination, mais les données et métadonnées de l'objet ne sont pas copiées.

Rubriques

Création d'un dossier

Cette section décrit comment utiliser la console Simple Storage Service (Amazon S3) pour créer un dossier.

> **Important**
>
> Si votre politique de compartiment empêche le chargement d'objets dans ce compartiment sans balises, métadonnées ni bénéficiaires de liste de contrôle d'accès (ACL), vous ne pouvez pas créer de dossier à l'aide de la procédure suivante. Au lieu de cela, chargez un dossier vide et spécifiez les paramètres suivants dans la configuration de chargement.

Pour créer un dossier

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment dans lequel vous souhaitez créer un dossier.
4. Si votre politique de compartiment empêche le chargement d'objets vers ce compartiment sans chiffrement, vous devez choisir Enable (Activer) sous Server-side encryption (Chiffrement côté serveur).
5. Choisissez Créer un dossier.
6. Attribuez un nom au dossier (par exemple, **`favorite-pics`**). Ensuite, choisissez Créer un dossier.

Rendre les dossiers publics

Nous vous recommandons de bloquer tout l'accès public à vos dossiers et compartiments Amazon S3, à moins que vous ayez besoin spécifiquement d'un dossier ou compartiment public. Lorsque vous rendez public un dossier, quiconque sur Internet peut voir tous les objets qui sont regroupés dans ce dossier.

Dans la console Amazon S3, vous pouvez rendre public un dossier. Vous pouvez également rendre public un dossier en créant une stratégie de compartiment qui limite l'accès aux données par préfixe. Pour plus d'informations, consultez Identity and Access Management pour Amazon S3.

> **Warning**
>
> Une fois que vous avez rendu public un dossier dans la console Amazon S3, vous ne pouvez pas le rendre à nouveau privé. Au lieu de cela, vous devez définir des autorisations sur chaque objet individuel dans le dossier public de sorte que les objets n'aient pas d'accès public. Pour plus d'informations, consultez Configuration des listes ACL.

Rubriques

- Calcul de la taille d'un dossier
- Suppression de dossiers

Calcul de la taille d'un dossier

Cette section décrit comment utiliser la console Amazon S3 pour calculer la taille d'un dossier.

Pour calculer la taille d'un dossier

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment dans lequel votre dossier est stocké.
4. Dans la liste Objects (Objets), cochez la case à côté du nom du dossier.
5. Choisissez Actions, puis Calculate total size (Calculer la taille totale).

> **Note**
>
> Les informations du dossier (y compris la taille totale) ne seront plus disponibles une fois que vous aurez quitté la page. Vous devez recalculer la taille totale si vous souhaitez la voir à nouveau.

> **Important**
>
> - Lorsque vous utilisez l'action Calculate total size (Calculer la taille totale) sur des objets ou des dossiers spécifiés dans votre compartiment, Amazon S3 calcule le nombre total d'objets et la taille de stockage totale. Toutefois, les téléchargements partitionnés incomplets ou en cours et les versions précédentes ou anciennes ne sont pas inclus dans le calcul du nombre total d'objets ou de la taille totale. Cette action calcule uniquement le nombre total d'objets et la taille totale de la version actuelle ou la plus récente de chaque objet stocké dans le compartiment.
>
>   Par exemple, si votre compartiment contient deux versions d'un même objet, le calculateur de stockage d'Amazon S3 les compte comme un seul objet. Par conséquent, le nombre total d'objets calculé dans la console Amazon S3 peut différer du nombre d'objets indiqué dans S3 Storage Lens et du nombre indiqué par le CloudWatch métrique Amazon,`NumberOfObjects`. De même, la taille de stockage totale peut également

différer de la métrique de stockage total indiquée dans S3 Storage Lens et de la `BucketSizeBytes` métrique indiquée dans CloudWatch.

- Si le calcul de la taille totale d'un dossier volumineux prend trop de temps, pensez à utiliser Amazon S3 Inventory et Amazon S3 Select comme alternative. Tout d'abord, créez une configuration S3 Inventory pour inclure les métadonnées de taille pour chaque objet du dossier volumineux dans un rapport d'inventaire. La distribution du premier rapport S3 Inventory peut prendre jusqu'à 48 heures. Lorsque le rapport d'inventaire est publié, interrogez-le à l'aide d'une expression `SUM` S3 Select pour agréger les tailles des objets du dossier. Pour plus d'informations, consultez Configuration de l'inventaire à l'aide de la console S3 et Exemple de SUM.

Suppression de dossiers

Cette section explique comment utiliser la console Amazon S3 pour supprimer des dossiers d'un compartiment S3.

Pour plus d'informations sur les fonctionnalités et la tarification d'Amazon S3, consultez Amazon S3.

Pour supprimer des dossiers d'un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment à partir duquel vous souhaitez supprimer des dossiers.
3. Dans la liste Objets, cochez la case en regard des dossiers et objets que vous souhaitez supprimer.
4. Sélectionnez Delete (Supprimer).
5. Dans la page Delete objects (Supprimer des objets), vérifiez que les noms des dossiers que vous avez sélectionnés pour la suppression sont répertoriés.
6. Dans la zone Supprimer les objets, saisissez **`delete`**, et choisissez Supprimer les objets.

> **Warning**
>
> Cette action supprime tous les objets spécifiés. Lorsque vous supprimez des dossiers, attendez la fin de l'action de suppression pour ajouter de nouveaux objets au dossier. Dans le cas contraire, de nouveaux objets pourraient également être supprimés.

Affichage d'une présentation d'un objet dans la console Amazon S3

Vous pouvez utiliser la console Amazon S3 pour afficher une présentation d'un objet. La console fournit toutes les informations essentielles pour un objet dans un même emplacement.

Pour ouvrir la page de détails d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, choisissez le nom de l'objet pour lequel vous souhaitez obtenir une présentation.

   La page de détails de l'objet s'ouvre.
4. Pour télécharger l'objet, choisissez Actions d'objet, puis choisissez Télécharger. Pour copier le chemin d'accès de l'objet dans le presse-papiers, sous URL de l'objet, choisissez l'URL.
5. Si le contrôle de version est activé pour le compartiment, choisissez Versions pour afficher la liste de toutes les versions de l'objet.

   - Pour télécharger une version d'objet, cochez la case en regard de l'ID de version, choisissez Actions, puis choisissez Télécharger.
   - Pour supprimer une version d'objet, cochez la case en regard de l'ID de version, puis choisissez Supprimer.

   > **Important**
   >
   > Vous pouvez annuler la suppression d'un objet uniquement si celui-ci a été supprimé en tant que version la plus récente (version actuelle). Vous ne pouvez pas restaurer une version précédente d'un objet supprimé.

Affichage des propriétés d'un objet dans la console Amazon S3

Vous pouvez utiliser la console Amazon S3 pour afficher les propriétés d'un objet, y compris la classe de stockage, les paramètres de chiffrement, les étiquettes et les métadonnées.

Pour afficher les propriétés d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, choisissez le nom de l'objet pour lequel vous souhaitez afficher les propriétés.

   La Présentation de l'objet de votre objet s'ouvre. Vous pouvez faire défiler l'écran vers le bas pour afficher les propriétés de l'objet.
4. Sur la page Présentation de l'objet, vous pouvez configurer les propriétés suivantes pour l'objet.

> **Note**
>
> - Si vous modifiez les propriétés Storage Class (Classe de stockage), Encryption (Chiffrement) et Metadata (Métadonnées), un nouvel objet est créé pour remplacer l'ancien. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Le rôle qui modifie la propriété devient également le propriétaire du nouvel objet ou (version de l'objet).
> - Si vous modifiez les propriétés de classe de stockage, de chiffrement ou de métadonnées d'un objet doté de balises définies par l'utilisateur, vous devez disposer de cette `s3:GetObjectTagging` autorisation. Si vous modifiez ces propriétés pour un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`s3:GetObjectTagging`autorisation.
>
>   Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, ces propriétés de l'objet seront mises à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

a. Storage Class (Classe de stockage) – Chaque objet dans Amazon S3 possède une classe de stockage qui lui est associée. La classe de stockage que vous choisissez d'utiliser dépend de la fréquence à laquelle vous accédez à l'objet. La classe de stockage par défaut pour les objets S3 est STANDARD. Vous choisissez la classe de stockage à utiliser lors du chargement d'un objet. Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

   Pour modifier la classe de stockage après le chargement d'un objet, choisissez Classe de stockage. Choisissez la classe de stockage de votre choix, puis cliquez sur Enregistrer.

b. Paramètres de chiffrement côté serveur – Vous pouvez utiliser le chiffrement côté serveur pour chiffrer vos objets S3. Pour plus d'informations, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS) ou Spécification du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

c. Metadata (Métadonnées) – Chaque objet dans Amazon S3 possède un ensemble de paires nom-valeur qui représente ses métadonnées. Pour en savoir plus sur l'ajout de métadonnées à un objet S3, veuillez consulter Modification des métadonnées d'objet dans la console Amazon S3.

d. Balises – Vous catégorisez le stockage en ajoutant des balises à un objet S3. Pour plus d'informations, consultez Catégorisation de votre stockage à l'aide de balises.

e. Conservation et conservation légales du verrouillage des objets : vous pouvez empêcher la suppression d'un objet. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

Utilisation d'URL présignées

Vous pouvez utiliser des URL présignées pour accorder un accès limité dans le temps aux objets dans Amazon S3 sans mettre à jour votre stratégie de compartiment. Une URL présignée peut être saisie dans un navigateur ou utilisée par un programme pour charger un objet. Les informations d'identification utilisées par l'URL présignée sont celles de l' AWS utilisateur qui a généré l'URL.

Vous pouvez également utiliser des URL présignées pour permettre à quelqu'un de charger un objet spécifique dans votre compartiment Amazon S3. Cela permet un téléchargement sans qu'une autre partie ne doive disposer d'informations d'identification ou d'autorisations de AWS sécurité. Si un objet avec la même clé que celle spécifiée dans l'URL présignée existe déjà dans le compartiment, Amazon S3 remplace l'objet existant par l'objet chargé.

Vous pouvez utiliser l'URL présignée plusieurs fois, jusqu'à la date et l'heure d'expiration.

Lorsque vous créez une URL présignée, vous devez fournir vos informations d'identification de sécurité, puis spécifier les éléments suivants :

- Un compartiment Amazon S3
- Une clé d'objet (si le téléchargement de cet objet se fait dans votre compartiment Amazon S3, s'il s'agit du nom du fichier à charger)
- Une méthode HTTP (`GET` pour télécharger des objets ou `PUT` pour charger)
- Un intervalle de temps d'expiration

Pour l'heure, les URL présignées Amazon S3 ne prennent pas en charge l'utilisation des algorithmes de total de contrôle d'intégrité des données suivants (CRC32, CRC32C, SHA-1, SHA-256) lors du chargement d'objets. Pour vérifier l'intégrité de votre objet après son chargement, vous pouvez fournir un récapitulatif MD5 de l'objet lorsque vous le chargez avec une URL présignée. Pour en savoir plus sur l'intégrité des objets, consultez Vérification de l'intégrité des objets.

Rubriques

Utilisateurs habilités à créer une URL présignée

Toute personne qui possède des autorisations de sécurité valides peut créer une URL présignée. Mais pour que l'utilisateur puisse accéder correctement à un objet, l'URL présignée doit être créée par une personne qui possède l'autorisation d'effectuer l'opération sur laquelle l'URL présignée est basée.

Les informations d'identification que vous pouvez utiliser pour créer une URL présignée sont les suivantes :

- Profil d'instance IAM : valide pendant 6 heures.

- AWS Security Token Service : valide jusqu'à un maximum de 36 heures en cas de signature avec des informations d'identification de sécurité à long terme ou pendant la durée de validité des informations d'identification temporaires, selon l'élément qui se termine en premier.
- Utilisateur IAM — Valable jusqu'à 7 jours lorsque vous utilisez AWS Signature version 4.

  Afin de créer une URL présignée valide pendant 7 jours maximum, commencez par déléguer des informations d'identification d'utilisateur IAM (clé d'accès et clé secrète) à la méthode que vous utilisez pour créer l'URL présignée.

> **Note**
>
> Si vous avez créé une URL présignée à l'aide d'informations d'identification temporaires, l'URL expire quand les informations d'identification expirent. En général, une URL présignée expire lorsque les informations d'identification que vous avez utilisées pour la créer sont révoquées, supprimées ou désactivées. Cela est vrai même si l'URL a été créée avec une date d'expiration ultérieure. Pour connaître la durée de vie des informations d'identification de sécurité temporaires, consultez la section Comparaison des opérations AWS STS d'API dans le guide de l'utilisateur IAM.

Délai d'expiration pour les URL présignées

Une URL présignée reste valide pendant la période spécifiée lors de sa génération. Si vous créez une URL présignée à l'aide de la console Amazon S3, le délai d'expiration peut être défini entre 1 minute et 12 heures. Si vous utilisez les AWS SDK AWS CLI ou, le délai d'expiration peut être fixé à 7 jours.

Si vous avez créé une URL présignée à l'aide d'un jeton temporaire, l'URL expire lorsque le jeton expire. En général, une URL présignée expire lorsque les informations d'identification que vous avez utilisées pour la créer sont révoquées, supprimées ou désactivées. Cela est vrai même si l'URL a été créée avec une date d'expiration ultérieure. Pour plus d'informations sur la manière dont les informations d'identification que vous utilisez affectent le délai d'expiration, consultez Utilisateurs habilités à créer une URL présignée.

Amazon S3 vérifie la date et l'heure d'expiration d'une URL signée au moment de la requête HTTP. Par exemple, si un client commence à télécharger un fichier volumineux immédiatement avant la date d'expiration, le téléchargement continue même si la date d'expiration intervient pendant le

téléchargement. Cependant, si la connexion est perdue et que le client essaie de redémarrer le téléchargement une fois la date d'expiration passée, le téléchargement échoue.

Limitation des capacités des URL présignées

Les capacités d'une URL présignée sont limitées par les autorisations de l'utilisateur qui l'a créée. En résumé, les URL présignées correspondent à des jetons porteurs qui donnent accès à ceux qui les possèdent. À ce titre, nous vous recommandons de les protéger de manière appropriée. Voici quelques méthodes que vous pouvez utiliser pour restreindre l'utilisation de vos URL présignées.

AWS Version 4 de la signature (SigV4)

Pour imposer un comportement spécifique lorsque les requêtes d'URL présignées sont authentifiées à l'aide d' AWS Signature Version 4 (SigV4), vous pouvez utiliser les clés de condition dans les stratégies de compartiment et les stratégies de point d'accès. Par exemple, la stratégie de compartiment suivante utilise la condition `s3:signatureAge` pour refuser toute demande d'URL présignée Amazon S3 sur les objets du compartiment *`example-s3-bucket1`* si la signature date de plus de 10 minutes. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Deny a presigned URL request if the signature is more than 10 min
 old",
         "Effect": "Deny",
         "Principal": {"AWS":"*"},
         "Action": "s3:*",
         "Resource": "arn:aws:s3:::example-s3-bucket1/*",
         "Condition": {
            "NumericGreaterThan": {
               "s3:signatureAge": 600000
            }
         }
      }
   ]
}
```

Pour plus d'informations sur les clés de politique relatives à AWS la version 4 de Signature, consultez AWS la section Authentification de signature version 4 dans le manuel Amazon Simple Storage Service API Reference.

Restriction de chemin réseau

Si vous souhaitez restreindre l'utilisation d'URL présignées et tous les accès Amazon S3 à des chemins réseau particuliers, vous pouvez rédiger des politiques AWS Identity and Access Management (IAM). Vous pouvez définir ces politiques sur le principal IAM qui effectue l'appel, le compartiment Simple Storage Service (Amazon S3) ou les deux.

Une restriction de chemin réseau sur le principal IAM exige que l'utilisateur de ces informations d'identification effectue des requêtes à partir du réseau spécifié. Une restriction sur le compartiment ou le point d'accès nécessite que toutes les requêtes adressées à cette ressource proviennent du réseau spécifié. Ces restrictions s'appliquent également hors du scénario des URL présignées.

La clé de condition globale IAM que vous utilisez dépend du type de point de terminaison. Si vous utilisez le point de terminaison public pour Amazon S3, utilisez `aws:SourceIp`. Si vous utilisez un point de terminaison de cloud privé virtuel (VPC) pour Amazon S3, utilisez `aws:SourceVpc` ou `aws:SourceVpce`.

La déclaration de politique IAM suivante exige que le principal AWS n'accède qu'à partir de la plage réseau spécifiée. Avec cette déclaration de stratégie, tous les accès doivent provenir de cette plage, Cela inclut lorsqu'une personne utilise une URL présignée pour Amazon S3. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Pour d'autres exemples de politiques de compartiment qui utilisent la clé de condition `aws:SourceIp` AWS globale pour restreindre l'accès à un compartiment Amazon S3 à une plage réseau spécifique, consultezGestion de l'accès en fonction d'adresses IP spécifiques.

Partage d'objets à l'aide d'URL présignées

Par défaut, tous les objets Amazon S3 sont privés, seul le propriétaire de l'objet a l'autorisation d'y accéder. Toutefois, le propriétaire de l'objet peut partager des objets avec d'autres personnes en créant une URL présignée. Une URL présignée utilise des informations d'identification de sécurité pour accorder une autorisation limitée dans le temps pour télécharger des objets. L'URL peut être saisie dans un navigateur ou utilisée par un programme pour télécharger l'objet. Les informations d'identification utilisées par l'URL présignée sont celles de l' AWS utilisateur qui a généré l'URL.

Pour des informations générales sur les URL présignées, consultez Utilisation d'URL présignées.

Vous pouvez créer une URL présignée pour partager un objet sans écrire de code en utilisant la console Amazon S3, AWS Explorer pour Visual Studio (Windows) ou AWS Toolkit for Visual Studio Code. Vous pouvez également générer une URL présignée par programmation en utilisant le AWS Command Line Interface (AWS CLI) ou les SDK. AWS

Utilisation de la console S3

Vous pouvez utiliser la console Amazon S3 afin de générer une URL présignée pour partager un objet en suivant ces étapes. Dans la console, le délai d'expiration maximal d'une URL présignée est de 12 heures à compter de la création.

Pour générer une URL présignée à l'aide de la console Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet pour lequel vous souhaitez une URL pré-signée.
4. Dans la liste Objects (Objets), sélectionnez l'objet pour lequel vous souhaitez créer une URL présignée.
5. Dans le menu Actions d'objet, choisissez Partager avec une URL présignée.
6. Spécifiez la durée de validité souhaitée pour l'URL présignée.
7. Choisissez Create presigned URL (Créer une URL présignée).
8. Lorsqu'une confirmation apparaît, l'URL est automatiquement copiée dans votre presse-papier. Un bouton s'affiche pour copier l'URL présignée, si vous devez la copier à nouveau.

À l'aide du AWS CLI

L'exemple de AWS CLI commande suivant génère une URL présignée pour partager un objet depuis un compartiment Amazon S3. Lorsque vous utilisez le AWS CLI, le délai d'expiration maximal d'une URL présignée est de 7 jours à compter de sa création. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
aws s3 presign s3://example-s3-bucket1/mydoc.txt --expires-in 604800
```

> **Note**
>
> Pour tous Régions AWS ceux lancés après le 20 mars 2019, vous devez spécifier le `endpoint-url` et `Région AWS` avec la demande. Pour voir la liste complète des régions et points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Référence générale AWS .

```
aws s3 presign s3://example-s3-bucket1/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Pour plus d'informations, consultez la section presign dans la référence des commandes AWS CLI .

Utilisation des AWS SDK

Pour des exemples d'utilisation des AWS SDK pour générer une URL présignée pour partager un objet, consultez Créer une URL présignée pour Amazon S3 à l'aide d'un SDK. AWS

Lorsque vous utilisez les AWS SDK pour générer une URL présignée, le délai d'expiration maximal est de 7 jours à compter de la date de création.

> **Note**
>
> Pour tous Régions AWS ceux lancés après le 20 mars 2019, vous devez spécifier le `endpoint-url` et `Région AWS` avec la demande. Pour voir la liste complète des régions et points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Référence générale AWS .

> **Note**
>
> Lorsque vous utilisez les AWS SDK, l'attribut Tagging doit être un en-tête et non un paramètre de requête. Tous les autres attributs peuvent être transmis en tant que paramètre pour l'URL présignée.

À l'aide du AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> À l'heure actuelle, Visual Studio pour Mac AWS Toolkit for Visual Studio n'est pas compatible.

1. Installez le AWS Toolkit for Visual Studio en suivant les instructions ci-dessous, Installation et configuration du Toolkit for Visual Studio dans le guide de AWS Toolkit for Visual Studio l'utilisateur.
2. Connectez-vous à AWS en suivant les étapes ci-dessous, section Connexion à AWS dans le guide de AWS Toolkit for Visual Studio l'utilisateur.
3. Dans le panneau latéral gauche intitulé AWS Explorateur, double-cliquez sur le compartiment contenant votre objet.
4. Cliquez avec le bouton droit sur l'objet pour lequel vous souhaitez générer une URL présignée et sélectionnez Créer une URL pré-signée... .
5. Dans la fenêtre contextuelle, définissez la date et l'heure d'expiration de votre URL présignée.
6. La clé d'objet doit être préremplie en fonction de l'objet que vous avez sélectionné.
7. Choisissez GET pour spécifier que cette URL présignée est utilisée pour télécharger un objet.
8. Cliquez sur le bouton Générer.
9. Pour copier l'URL dans le presse-papiers, choisissez Copier.
10. Pour utiliser l'URL présignée générée, collez-la dans n'importe quel navigateur.

En utilisant AWS Toolkit for Visual Studio Code

Si vous utilisez Visual Studio Code, vous pouvez générer une URL pré-signée pour partager un objet sans écrire de code grâce à AWS Toolkit for Visual Studio Code. Pour des informations générales, consultez AWS Toolkit for Visual Studio Code dans le Guide de l'utilisateur AWS Toolkit for Visual Studio Code .

Pour obtenir des instructions sur l'installation du AWS Toolkit for Visual Studio Code, reportez-vous à la section Installation du AWS Toolkit for Visual Studio Code dans le guide de AWS Toolkit for Visual Studio Code l'utilisateur.

1. Connectez-vous à AWS en suivant les étapes ci-dessous, section Connexion à AWS Toolkit for Visual Studio Code dans le guide de AWS Toolkit for Visual Studio Code l'utilisateur.
2. Sélectionnez le AWS logo sur le panneau de gauche dans Visual Studio Code.
3. Sous EXPLORER, sélectionnez S3.
4. Choisissez un compartiment et un fichier, puis ouvrez le menu contextuel (clic droit).
5. Choisissez Générer une URL présignée, puis définissez le délai d'expiration (en minutes).
6. Appuyez sur Entrée et l'URL présignée est copiée dans votre presse-papiers.

Chargement d'objets à l'aide d'URL présignées

Vous pouvez utiliser des URL présignées pour permettre à quelqu'un de charger un objet dans votre compartiment Amazon S3. L'utilisation d'une URL présignée permettra un téléchargement sans qu'une autre partie n'ait besoin d'informations d'identification ou d'autorisations AWS de sécurité. Une URL présignée est limitée par les autorisations de l'utilisateur qui l'a créée. Cela signifie que si vous recevez une URL présignée pour charger un objet, vous pouvez le charger uniquement si le créateur de l'URL dispose des autorisations nécessaires pour charger cet objet.

Lorsqu'une personne utilise l'URL pour charger un objet, Amazon S3 crée l'objet dans le compartiment spécifié. Si un objet avec la même clé que celle spécifiée dans l'URL présignée existe déjà dans le compartiment, Simple Storage Service (Amazon S3) remplace l'objet existant par l'objet chargé. Après le chargement, le propriétaire du compartiment devient propriétaire de l'objet.

Pour des informations générales sur les URL présignées, consultez Utilisation d'URL présignées.

Vous pouvez créer une URL présignée pour le chargement d'un objet sans écrire de code grâce à AWS Explorer for Visual Studio. Vous pouvez également générer une URL présignée par programmation à l'aide des kits SDK AWS .

À l'aide du AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> À l'heure actuelle, Visual Studio pour Mac AWS Toolkit for Visual Studio n'est pas compatible.

1. Installez le AWS Toolkit for Visual Studio en suivant les instructions ci-dessous, Installation et configuration du Toolkit for Visual Studio dans le guide de AWS Toolkit for Visual Studio l'utilisateur.
2. Connectez-vous à AWS en suivant les étapes ci-dessous, section Connexion à AWS dans le guide de AWS Toolkit for Visual Studio l'utilisateur.
3. Dans le panneau de gauche intitulé AWS Explorateur, cliquez avec le bouton droit sur le compartiment dans lequel vous souhaitez charger un objet.
4. Choisissez Créer une URL pré-signée... .
5. Dans la fenêtre contextuelle, définissez la date et l'heure d'expiration de votre URL présignée.
6. Pour Object Key, définissez le nom du fichier à télécharger. Le fichier que vous chargez doit correspondre exactement à ce nom. Si un objet avec la même clé d'objet existe déjà dans le compartiment, Amazon S3 remplacera l'objet existant par l'objet récemment chargé.
7. Choisissez PUT pour spécifier que cette URL présignée est utilisée pour charger un objet.
8. Cliquez sur le bouton Générer.
9. Pour copier l'URL dans le presse-papiers, choisissez Copier.
10. Pour utiliser cette URL, vous pouvez envoyer une demande PUT avec la commande `curl`. Incluez le chemin complet de votre fichier et l'URL présignée elle-même.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Utilisation des AWS SDK

Pour des exemples d'utilisation des AWS SDK pour générer une URL présignée pour le téléchargement d'un objet, consultez Créer une URL présignée pour Amazon S3 à l'aide d'un SDK. AWS

Lorsque vous utilisez les AWS SDK pour générer une URL présignée, le délai d'expiration maximal est de 7 jours à compter de la date de création.

> **Note**
>
> Pour tous Régions AWS ceux lancés après le 20 mars 2019, vous devez spécifier le `endpoint-url` et `Région AWS` avec la demande. Pour voir la liste complète des régions

et points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Référence générale AWS .

Transformation d'objets avec S3 Object Lambda

Amazon S3 Object Lambda vous donne la possibilité d'ajouter votre propre code aux requêtes Amazon S3 `GET`, `LIST` et `HEAD` afin de modifier et de traiter les données lorsqu'elles sont renvoyées vers une application. Vous pouvez utiliser du code personnalisé pour modifier les données renvoyées par les demandes S3 `GET` afin de filtrer les lignes, de redimensionner les images et les filigranes de manière dynamique, de supprimer des données confidentielles et plus encore. Vous pouvez également utiliser S3 Object Lambda pour modifier la sortie des requêtes S3 `LIST` afin de créer une vue personnalisée de tous les objets d'un compartiment et des requêtes S3 `HEAD` pour modifier les métadonnées des objets, telles que le nom et la taille des objets. Vous pouvez utiliser S3 Object Lambda comme origine pour votre CloudFront distribution Amazon afin de personnaliser les données pour les utilisateurs finaux, par exemple en redimensionnant automatiquement les images, en transcodant d'anciens formats (comme de JPEG en WebP) ou en supprimant des métadonnées. Pour plus d'informations, consultez le billet de AWS blog Use Amazon S3 Object Lambda with Amazon. CloudFront Alimenté par les fonctions AWS Lambda, votre code s'exécute sur une infrastructure entièrement gérée par. AWS S3 Object Lambda réduit le besoin de créer et de stocker des copies dérivées de vos données ou d'exécuter des proxys, tout cela sans avoir à modifier vos applications.

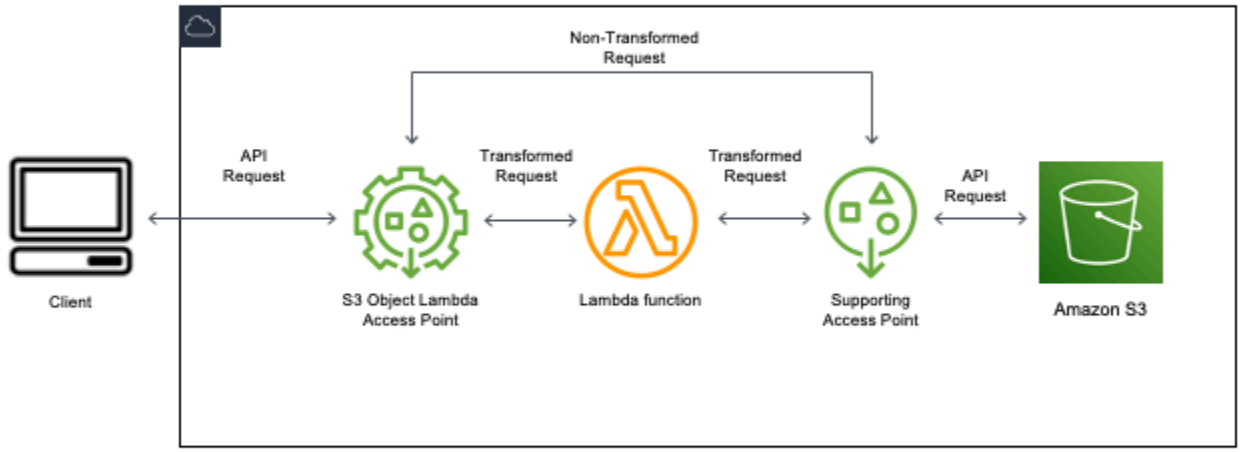
Fonctionnement de S3 Object Lambda

S3 Object Lambda utilise des AWS Lambda fonctions pour traiter automatiquement la sortie des requêtes S3 ou `GET` `HEAD` requêtes `LIST` standard. AWS Lambda est un service de calcul sans serveur qui exécute du code défini par le client sans nécessiter la gestion des ressources informatiques sous-jacentes. Vous pouvez créer et exécuter vos propres fonctions Lambda personnalisées afin d'ajuster la transformation des données à vos cas d'utilisation spécifiques.

Après avoir configuré une fonction Lambda, vous l'attachez à un point de terminaison de service S3 Object Lambda (connu sous le nom de point d'accès Object Lambda). Le point d'accès Object Lambda utilise un point d'accès S3 standard, appelé point d'accès de prise en charge, pour accéder à Amazon S3.

Lorsque vous envoyez une demande à votre point d'accès Object Lambda, Amazon S3 appelle automatiquement votre fonction Lambda. Toutes les données récupérées à l'aide d'une demande S3

`GET`, `LIST` ou `HEAD` via le point de terminaison Object Lambda renvoient un résultat transformé à l'application. Toutes les autres demandes sont traitées comme d'habitude, comme illustré dans le diagramme suivant.



Les rubriques de cette section décrivent comment utiliser S3 Object Lambda.

Rubriques

- Création de points d'accès Object Lambda
- Utilisation des points d'accès Amazon S3 Object Lambda
- Considérations sur la sécurité pour les points d'accès S3 Object Lambda
- Écriture de fonctions Lambda pour les points d'accès S3 Object Lambda
- Utilisation de AWS fonctions Lambda intégrées
- Bonnes pratiques et directives pour S3 Object Lambda
- Didacticiels S3 Object Lambda
- Débogage de S3 Object Lambda

Création de points d'accès Object Lambda

Un point d'accès Object Lambda est associé à exactement un point d'accès standard et donc à un compartiment Amazon S3. Pour créer un point d'accès Object Lambda, vous avez besoin des ressources suivantes :

- Un compartiment Amazon S3. Pour plus d'informations sur la création de compartiments, consultez the section called “Créer un compartiment”.
- Un point d'accès S3 standard. Lorsque vous utilisez des points d'accès Object Lambda, ce point d'accès standard est appelé point d'accès de prise en charge. Pour obtenir des informations sur la création de points d'accès standard, consultez the section called “Création de points d'accès”.
- Une AWS Lambda fonction. Vous pouvez créer votre propre fonction Lambda ou utiliser une fonction prédéfinie. Pour plus d'informations sur la création de fonctions Lambda, consultez the section called “Écriture de fonctions Lambda”. Pour plus d'informations sur les fonctions prédéfinies, consultez Utilisation de AWS fonctions Lambda intégrées.
- (Facultatif) Une politique AWS Identity and Access Management (IAM). Les points d'accès Amazon S3 prennent en charge les politiques de ressources IAM que vous pouvez utiliser pour contrôler l'utilisation des points d'accès par ressource, utilisateur ou d'autres conditions. Pour plus d'informations sur la création de telles politiques, consultez the section called “Configuration des stratégies IAM”.

Les sections suivantes décrivent comment créer un point d'accès Object Lambda en utilisant :

- Le AWS Management Console
- Le AWS Command Line Interface (AWS CLI)
- Un AWS CloudFormation modèle
- Le AWS Cloud Development Kit (AWS CDK)

Pour obtenir des informations sur la création d'un point d'accès Object Lambda à l'aide de l'API REST, consultez `CreateAccessPointForObjectLambda` dans la Référence d'API Amazon Simple Storage Service.

Créer un point d'accès Object Lambda

Utilisez l'une des procédures suivantes pour créer votre point d'accès Object Lambda.

Utilisation de la console S3

Pour créer un point d'accès Object Lambda à l'aide de la console

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation, choisissez le nom du fichier actuellement affiché Région AWS. Ensuite, choisissez la région vers laquelle vous souhaitez passer.
3. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
4. Dans la page Points d'accès Object Lambda), choisissez Créer un point d'accès Object Lambda).
5. Pour Object Lambda Access Point name (Nom du point d'accès Object Lambda), saisissez le nom que vous souhaitez utiliser pour le point d'accès.

   Comme pour les points d'accès standard, des règles s'appliquent à l'attribution de noms pour des points d'accès Object Lambda. Pour plus d'informations, consultez Règles relatives à l'attribution de noms pour les points d'accès Amazon S3.
6. Pour Supporting Access Point (Point d'accès de prise en charge), saisissez le point d'accès standard à utiliser ou accédez-y. Le point d'accès doit se trouver dans le même Région AWS emplacement que les objets que vous souhaitez transformer. Pour obtenir des informations sur la création de points d'accès standard, consultez the section called “Création de points d'accès”.
7. Sous Configuration de transformation, vous pouvez ajouter une fonction qui transforme vos données pour votre point d'accès Object Lambda. Effectuez l'une des actions suivantes :

   - Si vous avez déjà une AWS Lambda fonction dans votre compte, vous pouvez la sélectionner sous Invoke Lambda function. Vous pouvez saisir ici l'Amazon Resource Name (ARN) d'une fonction Lambda dans votre Compte AWS ou choisir une fonction Lambda dans le menu déroulant.
   - Si vous souhaitez utiliser une fonction AWS intégrée, choisissez le nom de la fonction sous fonction AWS construite et sélectionnez Créer une fonction Lambda. Cela vous mènera à la console Lambda où vous pourrez déployer une fonction intégrée dans votre. Compte AWS Pour plus d'informations sur les fonctions définies, consultez Utilisation de AWS fonctions Lambda intégrées.

   Sous S3 APIs (API S3), choisissez une ou plusieurs opérations d'API à appeler. Pour chaque API sélectionnée, vous devez spécifier une fonction Lambda à appeler.

8. (Facultatif) Sous Payload (Charge utile), ajoutez le texte JSON à fournir en entrée à la fonction Lambda. Vous pouvez configurer des charges utiles avec différents paramètres pour différents points d'accès Object Lambda qui invoquent la même fonction Lambda, augmentant ainsi la flexibilité de votre fonction Lambda.

   > **Important**
   >
   > Lorsque vous utilisez des points d'accès Object Lambda, vérifiez que la charge utile ne contient pas d'informations confidentielles.

9. (Facultatif) Pour Range and part number (Plage et numéro de partie), vous devez activer cette option pour traiter les requêtes `GET` et `HEAD` avec des en-têtes de plage et de numéro de partie. L'activation de cette option confirme que votre fonction Lambda est capable de reconnaître et de traiter ces demandes. Pour plus d'informations sur les en-têtes de plage et les numéros de partie, voir Utilisation des en-têtes Range et partNumber.
10. (Facultatif) Pour Métriques de demande), choisissez Activer ou Désactiver pour ajouter la surveillance Amazon S3 à votre point d'accès Object Lambda. Les statistiques relatives aux demandes sont facturées au CloudWatch tarif standard d'Amazon.
11. (Facultatif) Sous Object Lambda Access Point policy (Politique de points d'accès Object Lambda), définissez une politique de ressources. Les politiques de ressources accordent des autorisations pour le point d'accès Object Lambda spécifié et peuvent contrôler l'utilisation du point d'accès par ressource, utilisateur ou d'autres conditions. Pour plus d'informations sur les politiques de ressources d'un point d'accès Object Lambda, consultez Configuration des politiques IAM pour les points d'accès Object Lambda.
12. Sous Block Public Access settings for this Object Lambda Access Point (Paramètres de blocage d'accès public pour ce point d'accès Object Lambda), sélectionnez les paramètres de blocage d'accès public que vous voulez appliquer. Tous les paramètres de blocage d'accès public sont activés par défaut pour les nouveaux points d'accès Object Lambda et nous vous recommandons de laisser les paramètres par défaut activés. Actuellement, Amazon S3 ne prend pas en charge la modification des paramètres de blocage d'accès public d'un point d'accès Object Lambda après la création de ce point d'accès Object Lambda.

    Pour plus d'informations sur l'utilisation du blocage de l'accès public Amazon S3, consultez Gestion de l'accès public aux points d'accès.
13. Choisissez Create Object Lambda Access Point (Créer un point d'accès Object Lambda).

À l'aide du AWS CLI

Pour créer un point d'accès Object Lambda à l'aide d'un modèle AWS CloudFormation

> **Note**
>
> Pour utiliser les commandes suivantes, remplacez *`user input placeholders`* par vos propres informations.

1. Téléchargez le package de déploiement des AWS Lambda fonctions `s3objectlambda_deployment_package.zip` dans la configuration par défaut de S3 Object Lambda.
2. Exécutez la commande `put-object` suivante pour charger le package dans un compartiment Amazon S3.

```
aws s3api put-object --bucket Amazon S3 bucket name --key
 s3objectlambda_deployment_package.zip --body release/
s3objectlambda_deployment_package.zip
```

3. Téléchargez le AWS CloudFormation modèle dans la `s3objectlambda_defaultconfig.yaml` configuration par défaut de S3 Object Lambda.
4. Exécutez la commande `deploy` suivante pour déployer le modèle dans votre Compte AWS.

```
aws cloudformation deploy --template-file s3objectlambda_defaultconfig.yaml \
 --stack-name AWS CloudFormation stack name \
 --parameter-overrides ObjectLambdaAccessPointName=Object Lambda Access Point name
 \
  SupportingAccessPointName=Amazon S3 access point S3BucketName=Amazon S3 bucket \
  LambdaFunctionS3BucketName=Amazon S3 bucket containing your Lambda package \
  LambdaFunctionS3Key=Lambda object key LambdaFunctionS3ObjectVersion=Lambda object
 version \
  LambdaFunctionRuntime=Lambda function runtime --capabilities capability_IAM
```

Vous pouvez configurer ce AWS CloudFormation modèle pour appeler Lambda pour `GETHEAD`, et les opérations d'`LIST`API. Pour plus d'informations sur la modification de la configuration par défaut du modèle, consultez the section called “Automatisez la configuration de S3 Object Lambda avec AWS CloudFormation”.

Pour créer un point d'accès Object Lambda à l'aide du AWS CLI

> **Note**
>
> Pour utiliser les commandes suivantes, remplacez *`user input placeholders`* par vos propres informations.

L'exemple suivant crée un point d'accès Object Lambda nommé *`my-object-lambda-ap`* pour le compartiment *`DOC-EXAMPLE-BUCKET1`* dans le compte *`111122223333`*. Cet exemple suppose qu'un point d'accès standard nommé *`example-ap`* a déjà été créé. Pour plus d'informations sur la création d'un point d'accès standard, veuillez consulter the section called “Création de points d'accès”.

Cet exemple utilise la fonction AWS prédéfinie. `decompress` Pour plus d’informations sur les fonctions prédéfinies, consultez the section called “Utilisation des fonctions AWS intégrées”.

1. Créez un compartiment. Dans cet exemple, nous allons utiliser *`DOC-EXAMPLE-BUCKET1`*. Pour plus d’informations sur la création de compartiments, consultez the section called “Créer un compartiment”.
2. Créez un point d'accès standard et attachez-le à votre compartiment. Dans cet exemple, nous allons utiliser *`example-ap`*. Pour obtenir des informations sur la création de points d'accès standard, consultez the section called “Création de points d'accès”.
3. Effectuez l’une des actions suivantes :

   - Créez une fonction Lambda dans votre compte que vous souhaitez utiliser pour transformer votre objet Simple Storage Service (Amazon S3). Pour plus d'informations sur la création de fonctions Lambda, consultez the section called “Écriture de fonctions Lambda”. Pour utiliser votre fonction personnalisée avec le AWS CLI, consultez la section Utilisation de Lambda avec le AWS CLI dans le guide du AWS Lambda développeur.
   - Utilisez une fonction AWS Lambda prédéfinie. Pour plus d’informations sur les fonctions prédéfinies, consultez Utilisation de AWS fonctions Lambda intégrées.
4. Créez un fichier de configuration JSON nommé `my-olap-configuration.json`. Dans cette configuration, fournissez le point d'accès de prise en charge et l'Amazon Resource Name (ARN) de la fonction Lambda que vous avez créée au cours des étapes précédentes ou l'ARN de la fonction prédéfinie que vous utilisez.

Example

```
{
    "SupportingAccessPoint" : "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
    "TransformationConfigurations": [{
        "Actions" : ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
        "ContentTransformation" : {
            "AwsLambda": {
                "FunctionPayload" : "{\"compressionType\":\"gzip\"}",
                "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/
compress"
            }
        }
    }]
}
```

5. Exécutez la commande `create-access-point-for-object-lambda` pour créer votre point d'accès Object Lambda.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-ap --configuration file://my-olap-configuration.json
```

6. (Facultatif) Créez un fichier de stratégie JSON nommé `my-olap-policy.json`.

   L'ajout d'une politique de ressources de points d'accès Object Lambda peut contrôler l'utilisation du point d'accès par ressource, utilisateur ou d'autres conditions. Cette politique de ressources accorde l'autorisation `GetObject` pour le compte *444455556666* au point d'accès Object Lambda spécifié.

   Example

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "Grant account 444455556666 GetObject access",
            "Effect": "Allow",
            "Action": "s3-object-lambda:GetObject",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Resource": "your-object-lambda-access-point-arn"
        }
    ]
}
```

7. (Facultatif) Exécutez la commande `put-access-point-policy-for-object-lambda` pour définir votre politique de ressources.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333
 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

8. (Facultatif) Spécifiez une charge utile.

   Une charge utile est un JSON facultatif que vous pouvez fournir à votre AWS Lambda fonction en entrée. Vous pouvez configurer des charges utiles avec différents paramètres pour différents points d'accès Object Lambda qui invoquent la même fonction Lambda, augmentant ainsi la flexibilité de votre fonction Lambda.

   La configuration suivante du point d'accès Object Lambda affiche une charge utile avec deux paramètres.

```
{
 "SupportingAccessPoint": "AccessPointArn",
 "CloudWatchMetricsEnabled": false,
 "TransformationConfigurations": [{
  "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
   "AwsLambda": {
    "FunctionArn": "FunctionArn",
    "FunctionPayload": "{\"res-x\": \"100\",\"res-y\": \"100\"}"
   }
  }
 }]
}
```

   La configuration de point d'accès Object Lambda suivante montre une charge utile avec un seul paramètre et avec `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` et `HeadObject-PartNumber` activés.

```
{
    "SupportingAccessPoint":"AccessPointArn",
    "CloudWatchMetricsEnabled": false,
    "AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber", "HeadObject-
Range", "HeadObject-PartNumber"],
    "TransformationConfigurations": [{
        "Action": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
        "ContentTransformation": {
            "AwsLambda": {
                "FunctionArn":"FunctionArn",
                "FunctionPayload": "{\"compression-amount\": \"5\"}"
            }
        }
    }]
}
```

> **Important**
>
> Lorsque vous utilisez des points d'accès Object Lambda, vérifiez que la charge utile ne contient pas d'informations confidentielles.

Utilisation de la AWS CloudFormation console et du modèle

Vous pouvez créer un point d'accès Object Lambda à l'aide de la configuration par défaut fournie par Amazon S3. Vous pouvez télécharger un AWS CloudFormation modèle et le code source d'une fonction Lambda depuis le GitHub référentiel et déployer ces ressources pour configurer un point d'accès Object Lambda fonctionnel.

Pour plus d'informations sur la modification de la configuration par défaut du AWS CloudFormation modèle, consultezthe section called “Automatisez la configuration de S3 Object Lambda avec AWS CloudFormation”.

Pour plus d'informations sur la configuration des points d'accès Object Lambda AWS CloudFormation sans le modèle, consultez le guide `AWS::S3ObjectLambda::AccessPoint`de l'AWS CloudFormation utilisateur.

Pour charger le package de déploiement de votre fonction Lambda

1. Téléchargez le package de déploiement des AWS Lambda fonctions `s3objectlambda_deployment_package.zip` dans la configuration par défaut de S3 Object Lambda.
2. Chargez le package dans un compartiment Amazon S3.

Pour créer un point d'accès Object Lambda à l'aide de la console AWS CloudFormation

1. Téléchargez le AWS CloudFormation modèle dans la `s3objectlambda_defaultconfig.yaml` configuration par défaut de S3 Object Lambda.
2. Connectez-vous à la console AWS de gestion et ouvrez-la à l' AWS CloudFormation adresse https://console.aws.amazon.com/cloudformation.
3. Effectuez l'une des actions suivantes :
   - Si vous ne l'avez jamais utilisé AWS CloudFormation auparavant, sur la page d' AWS CloudFormation accueil, choisissez Create stack.
   - Si vous l'avez AWS CloudFormation déjà utilisé, dans le volet de navigation de gauche, choisissez Stacks. Choisissez Create stack (Créer une pile), puis With new resources (standard) (Avec de nouvelles ressources (standard)).
4. Pour Prerequisite - Prepare template (Prérequis – Préparer le modèle), choisissez Template is ready (Le modèle est prêt).
5. Pour Specify template (Spécifier le modèle), choisissez Upload a template file (Charger un fichier de modèle) et chargez `s3objectlambda_defaultconfig.yaml`.
6. Choisissez Next (Suivant).
7. Sur la page Specify stack details (Spécifier les détails de la pile), saisissez un nom pour la pile.
8. Dans la section Parameters (Paramètres), spécifiez les paramètres suivants, définis dans le modèle de pile :
   a. Pour CreateNewSupportingAccessPoint, effectuez l'une des opérations suivantes :
      - Si vous disposez déjà d'un point d'accès de prise en charge pour le compartiment S3 dans lequel vous avez chargé le modèle, choisissez false (faux).
      - Si vous souhaitez créer un nouveau point d'accès pour ce compartiment, choisissez true (vrai).

b. Pour EnableCloudWatchMonitoring, choisissez vrai ou faux, selon que vous souhaitez activer les métriques de CloudWatch demande et les alarmes Amazon.

c. (Facultatif) Pour LambdaFunctionPayload, ajoutez le texte JSON que vous souhaitez fournir à votre fonction Lambda en entrée. Vous pouvez configurer des charges utiles avec différents paramètres pour différents points d'accès Object Lambda qui invoquent la même fonction Lambda, augmentant ainsi la flexibilité de votre fonction Lambda.

> **Important**
>
> Lorsque vous utilisez des points d'accès Object Lambda, vérifiez que la charge utile ne contient pas d'informations confidentielles.

d. Pour LambdaFunctionRuntime, entrez votre environnement d'exécution préféré pour la fonction Lambda. Les options disponibles sont `nodejs14.x`, `python3.9` et `java11`.

e. Pour LambdaFunctionS3 BucketName, entrez le nom du compartiment Amazon S3 dans lequel vous avez chargé le package de déploiement.

f. Pour LambdaFunctionS3Key, entrez la clé d'objet Amazon S3 sur laquelle vous avez chargé le package de déploiement.

g. Pour LambdaFunctionS3 ObjectVersion, entrez la version de l'objet Amazon S3 dans laquelle vous avez chargé le package de déploiement.

h. Pour ObjectLambdaAccessPointName, entrez un nom pour votre point d'accès Object Lambda.

i. Pour S3 BucketName, entrez le nom du compartiment Amazon S3 qui sera associé à votre point d'accès Object Lambda.

j. Pour SupportingAccessPointName, entrez le nom de votre point d'accès compatible.

> **Note**
>
> Il s'agit d'un point d'accès associé au compartiment Amazon S3 que vous avez choisi à l'étape précédente. Si aucun point d'accès n'est associé à votre compartiment Amazon S3, vous pouvez configurer le modèle pour en créer un pour vous en choisissant true for CreateNewSupportingAccessPoint.

9. Choisissez Next (Suivant).

10. Sur la page Configurer les options de pile, choisissez Suivant.

Pour plus d'informations sur les paramètres facultatifs figurant sur cette page, consultez Définition des options des piles AWS CloudFormation dans le Guide de l'utilisateur AWS CloudFormation .

11. Sur la page Review (Vérification), choisissez Create stack (Créer une pile).

À l'aide du AWS Cloud Development Kit (AWS CDK)

Pour plus d'informations sur la configuration des points d'accès Object Lambda à l'aide de AWS CDK, consultez la section `AWS::S3ObjectLambda`Construct Library dans la référence de l'AWS Cloud Development Kit (AWS CDK) API.

Automatisez la configuration de S3 Object Lambda à l'aide d'un modèle CloudFormation

Vous pouvez utiliser un AWS CloudFormation modèle pour créer rapidement un point d'accès Amazon S3 Object Lambda. Le CloudFormation modèle crée automatiquement les ressources pertinentes, configure les rôles AWS Identity and Access Management (IAM) et met en place une AWS Lambda fonction qui gère automatiquement les demandes via le point d'accès Object Lambda. Ce CloudFormation modèle vous permet de mettre en œuvre les meilleures pratiques, d'améliorer votre niveau de sécurité et de réduire les erreurs causées par les processus manuels.

Ce GitHub référentiel contient le CloudFormation modèle et le code source de la fonction Lambda. Pour obtenir des instructions sur l'utilisation de ce modèle, consultez the section called “Création de points d'accès Object Lambda”.

La fonction Lambda fournie dans ce modèle n'exécute aucune transformation. Au lieu de cela, elle renvoie vos objets en l'état à partir de votre compartiment S3. Vous pouvez cloner la fonction et ajouter votre propre code de transformation pour modifier et traiter les données lorsqu'elles sont renvoyées vers une application. Pour plus d'informations sur la modification de votre fonction, consultez the section called “Modification de la fonction Lambda” et the section called “Écriture de fonctions Lambda”.

Modification du modèle.

Création d'un nouveau point d'accès de prise en charge

S3 Object Lambda utilise deux points d'accès, un point d'accès Object Lambda et un point d'accès S3 standard, appelé point d'accès de prise en charge. Lorsque vous effectuez une demande auprès

d'un point d'accès Object Lambda, S3 appelle Lambda en votre nom ou délègue la demande au point d'accès de prise en charge, en fonction de la configuration S3 Object Lambda. Vous pouvez créer un nouveau point d'accès de prise en charge en transmettant le paramètre suivant dans le cadre de la commande `aws cloudformation deploy` lors du déploiement du modèle.

```
CreateNewSupportingAccessPoint=true
```

Configuration d'une charge utile de fonction

Vous pouvez configurer une charge utile de manière à fournir des données supplémentaires à la fonction Lambda en transmettant le paramètre suivant dans le cadre de la commande `aws cloudformation deploy` lors du déploiement du modèle.

```
LambdaFunctionPayload="format=json"
```

Activation de la CloudWatch surveillance d'Amazon

Vous pouvez activer la CloudWatch surveillance en transmettant le paramètre suivant dans le cadre de la `aws cloudformation deploy` commande lors du déploiement du modèle.

```
EnableCloudWatchMonitoring=true
```

Ce paramètre active votre point d'accès Object Lambda pour les métriques de demande Amazon S3 et crée deux CloudWatch alarmes pour surveiller les erreurs côté client et côté serveur.

> **Note**
>
> CloudWatch L'utilisation d'Amazon entraînera des coûts supplémentaires. Pour plus d'informations sur les métriques de demande Amazon S3, consultez Surveillance et journalisation des points d'accès.
> Pour plus d'informations sur la tarification, consultez Tarification CloudWatch .

Configuration de la simultanéité provisionnée

Pour réduire la latence, vous pouvez configurer la simultanéité provisionnée pour la fonction Lambda qui soutient le point d'accès Object Lambda, en modifiant le modèle afin d'inclure les lignes suivantes sous `Resources`.

```
LambdaFunctionVersion:
      Type: AWS::Lambda::Version
      Properties:
        FunctionName: !Ref LambdaFunction
        ProvisionedConcurrencyConfig:
            ProvisionedConcurrentExecutions: Integer
```

> **Note**
>
> Des frais supplémentaires pour le provisionnement simultané vous seront facturés. Pour plus d'informations sur la simultanéité provisionnée, consultez Gestion de la simultanéité provisionnée Lambda dans le Guide du développeur AWS Lambda .
> Pour plus d'informations sur la tarification, consultez Tarification AWS Lambda.

Modification de la fonction Lambda

Modification des valeurs d'en-tête d'une requête **`GetObject`**

Par défaut, la fonction Lambda transfère tous les en-têtes, à l'exception de `Content-Length` et `ETag`, de la demande d'URL pré-signée au client `GetObject`. En fonction de votre code de transformation dans la fonction Lambda, vous pouvez choisir d'envoyer de nouvelles valeurs d'en-tête à la demande `GetObject` du client.

Vous pouvez mettre à jour votre fonction Lambda pour envoyer de nouvelles valeurs d'en-tête en les transmettant dans l'opération d'API `WriteGetObjectResponse`.

Par exemple, si votre fonction Lambda traduit du texte dans des objets Amazon S3 vers une autre langue, vous pouvez transmettre une nouvelle valeur dans l'en-tête `Content-Language`. Pour ce faire, vous pouvez modifier la fonction `writeResponse` comme suit :

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
```

```
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    ContentLanguage: 'my-new-language'
  }).promise();
}
```

Pour obtenir la liste complète des en-têtes pris en charge, consultez `WriteGetObjectResponse` dans la Référence d'API Amazon Simple Storage Service.

Renvoi des en-têtes de métadonnées

Vous pouvez mettre à jour votre fonction Lambda pour envoyer de nouvelles valeurs d'en-tête en les transmettant dans la demande d'opération d'API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest,
      'my-new-header': 'my-new-value'
    },
    ...headers
  }).promise();
}
```

Renvoi d'un nouveau code de statut

Vous pouvez renvoyer un code de statut personnalisé au client `GetObject` en transmettant ce code dans la demande d'opération d'API `WriteGetObjectResponse`.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);
```

```
  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

Pour obtenir la liste complète des codes de statut pris en charge, consultez `WriteGetObjectResponse` dans la Référence d'API Amazon Simple Storage Service.

Application des paramètres **`Range`** et **`partNumber`** à l'objet source

Par défaut, le point d'accès Object Lambda créé par le CloudFormation modèle peut gérer les paramètres `Range` et`partNumber`. La fonction Lambda applique la plage ou le numéro de partie demandé à l'objet transformé. Pour ce faire, la fonction doit télécharger l'objet entier et exécuter la transformation. Dans certains cas, vos plages d'objets transformés peuvent correspondre exactement à vos plages d'objets source. Cela signifie que la demande de plage d'octets A-B sur votre objet source et l'exécution de la transformation peuvent fournir le même résultat que la demande de l'objet entier, l'exécution de la transformation et le renvoi de la plage d'octets A-B sur l'objet transformé.

Dans ce cas, vous pouvez modifier l'implémentation de la fonction Lambda pour appliquer la plage ou le numéro de partie directement à l'objet source. Cette approche réduit la latence globale de la fonction ainsi que la mémoire requise. Pour plus d'informations, consultez the section called “Utilisation des en-têtes Range et partNumber”.

Désactivation de la gestion de **`Range`** et **`partNumber`**

Par défaut, le point d'accès Object Lambda créé par le CloudFormation modèle peut gérer les paramètres `Range` et`partNumber`. Si vous n'avez pas besoin de ce comportement, vous pouvez le désactiver en supprimant les lignes suivantes du modèle :

```
AllowedFeatures:
  - GetObject-Range
  - GetObject-PartNumber
```

```
  - HeadObject-Range
  - HeadObject-PartNumber
```

Transformation de large objets

Par défaut, la fonction Lambda traite l'ensemble de l'objet en mémoire avant de pouvoir commencer à diffuser la réponse vers S3 Object Lambda. Vous pouvez modifier la fonction pour diffuser la réponse au fur et à mesure qu'elle effectue la transformation. Cela permet de réduire la latence de transformation et la mémoire de la fonction Lambda. Pour obtenir un exemple de mise en œuvre, consultez Stream compressed content example (Exemple de diffusion de contenu compressé).

Utilisation des points d'accès Amazon S3 Object Lambda

Les demandes via des points d'accès Amazon S3 Object Lambda fonctionnent de la même manière que les demandes via d'autres points d'accès. Pour plus d'informations sur les demandes via un point d'accès, consultez Utilisation des points d'accès. Vous pouvez effectuer des demandes via les points d'accès Object Lambda à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI), des AWS SDK ou de l'API REST Amazon S3.

> **Important**
>
> Les Amazon Resource Names (ARN) des points d'accès Object Lambda utilisent le nom de service `s3-object-lambda`. Les ARN des points d'accès Object Lambda commencent donc par `arn:aws::s3-object-lambda`, et non par `arn:aws::s3`, qui est utilisé pour d'autres points d'accès.

Comment trouver l'ARN de votre point d'accès Object Lambda

Pour utiliser un point d'accès Object Lambda avec les AWS SDK AWS CLI OR, vous devez connaître le nom de ressource Amazon (ARN) du point d'accès Object Lambda. Les exemples suivants montrent comment trouver l'ARN d'un point d'accès Object Lambda à l'aide de la console Amazon S3 ou de l' AWS CLI.

Utilisation de la console S3

Pour trouver l'ARN de votre point d'accès Object Lambda avec la console

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.
3. Sélectionnez le bouton d'option en regard du point d'accès Object Lambda dont vous souhaitez copier l'ARN.
4. Choisissez Copier l'ARN.

À l'aide du AWS CLI

Pour trouver l'ARN de votre point d'accès Object Lambda à l'aide du AWS CLI

1. Pour récupérer la liste des points d'accès Object Lambda associés à votre Compte AWS, exécutez la commande suivante. Avant d'exécuter la commande, remplacez l'identifiant du compte *111122223333*par le Compte AWS vôtre.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Lisez la sortie de la commande pour trouver l'ARN du point d'accès Object Lambda que vous souhaitez utiliser. La sortie de la commande précédente doit ressembler à celle de l'exemple suivant.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/my-object-lambda-ap"
        },
        ...
    ]
}
```

Comment utiliser un alias de type compartiment pour votre point d'accès Object Lambda de compartiment S3

Quand vous créez un point d'accès Object Lambda, Amazon S3 génère automatiquement un alias unique pour votre point d'accès Object Lambda. Vous pouvez utiliser cet alias à la place d'un nom de compartiment Amazon S3 ou de l'Amazon Resource Name (ARN) du point d'accès Object Lambda dans une demande pour les opérations de plan de données de point d'accès. Pour obtenir la liste de ces opérations, veuillez consulter la page Compatibilité des points d'accès avec les AWS services.

Un nom d'alias de point d'accès Object Lambda est créé dans le même espace de noms qu'un compartiment Amazon S3. Ce nom d'alias est généré automatiquement et ne peut pas être modifié. Pour un point d'accès Object Lambda existant, un alias est automatiquement attribué. Un nom d'alias de point d'accès Object Lambda répond à toutes les exigences d'un nom de compartiment Amazon S3 valide et comprend les parties suivantes :

Object Lambda Access Point name prefix-*metadata*--ol-s3

> **Note**
>
> Le suffixe `--ol-s3` est réservé aux noms d'alias de point d'accès Object Lambda et ne peut pas être utilisé pour les noms de compartiment ou de point d'accès Object Lambda. Pour plus d'informations sur les règles d'attribution de noms des compartiments Amazon S3, consultez Règles de dénomination de compartiment.

Voici des exemples d'ARN et d'alias de point d'accès Object Lambda pour un point d'accès Object Lambda nommé *my-object-lambda-access-point* :

- ARN – arn:aws:s3-object-lambda:*region*:*account-id*:accesspoint/*my-object-lambda-access-point*
- Alias de point d'accès Object Lambda : *my-object-lambda-acc*-*1a4n8yjrb3kda96f67zwrwiiuse1a*--ol-s3

Lorsque vous utilisez un point d'accès Object Lambda, vous pouvez utiliser le nom d'alias du point d'accès Object Lambda sans d'importantes modifications de code.

Lorsque vous supprimez un point d'accès Object Lambda, le nom d'alias du point d'accès Object Lambda devient inactif et n'est plus provisionné.

Comment trouver l'alias de votre point d'accès Object Lambda

Utilisation de la console S3

Pour trouver l'alias de votre point d'accès Object Lambda avec la console

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Points d'accès Object Lambda.

3. Pour le point d'accès Object Lambda que vous souhaitez utiliser, copiez la valeur de l'alias du point d'accès Object Lambda.

À l'aide du AWS CLI

Lorsque vous créez un point d'accès Object Lambda, Amazon S3 génère automatiquement un nom d'alias de point d'accès Object Lambda, comme illustré dans l'exemple de commande suivant. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations. Pour plus d'informations sur la création d'un point d'accès Object Lambda à l'aide du AWS CLI, consultez. Pour créer un point d'accès Object Lambda à l'aide du AWS CLI

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-access-point --configuration file://my-olap-configuration.json
{
    "ObjectLambdaAccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-
access-point",
    "Alias": {
        "Value": "my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiiuse1a--ol-s3",
        "Status": "READY"
    }
}
```

Le nom d'alias du point d'accès Object Lambda généré comporte deux champs :

- Le champ `Value` est la valeur d'alias du point d'accès Object Lambda.
- Le champ `Status` est le statut de l'alias du point d'accès Object Lambda. Si le statut est `PROVISIONING`, Amazon S3 fournit l'alias du point d'accès Object Lambda, qui n'est pas encore prêt à être utilisé. Si le statut est `READY`, l'alias du point d'accès Object Lambda a été correctement provisionné et est prêt à être utilisé.

Pour plus d'informations sur le type de données `ObjectLambdaAccessPointAlias` dans l'API REST, consultez CreateAccessPointForObjectLambda et ObjectLambdaAccessPointAlias dans la Référence d'API Amazon Simple Storage Service.

Comment utiliser l'alias du point d'accès Object Lambda

Vous pouvez utiliser un alias de point d'accès Object Lambda plutôt qu'un nom de compartiment Amazon S3 pour les opérations répertoriées dans Compatibilité des points d'accès avec les AWS services.

L' AWS CLI exemple de `get-bucket-location` commande suivant utilise l'alias du point d'accès du compartiment pour renvoyer le nom dans Région AWS lequel se trouve le compartiment. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api get-bucket-location --bucket my-object-lambda-
acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3

{
    "LocationConstraint": "us-west-2"
}
```

Si l'alias du point d'accès Object Lambda indiqué dans une demande n'est pas valide, le code d'erreur `InvalidAccessPointAliasError` est renvoyé. Pour plus d'informations sur `InvalidAccessPointAliasError`, consultez List of Error Codes dans la Référence d'API Amazon Simple Storage Service.

Les limites d'un alias de point d'accès Object Lambda sont les mêmes que celles d'un alias de point d'accès. Pour plus d'informations sur les limites d'un alias de point d'accès, consultez Limites.

Considérations sur la sécurité pour les points d'accès S3 Object Lambda

Avec Amazon S3 Object Lambda, vous pouvez effectuer des transformations personnalisées sur les données lorsqu'elles quittent Amazon S3 en utilisant l'évolutivité et la flexibilité d'une AWS Lambda plate-forme de calcul. S3 et Lambda restent sécurisés par défaut, mais une attention particulière de l'auteur de fonction Lambda est nécessaire pour maintenir cette sécurité. S3 Object Lambda nécessite que tous les accès soient effectués par des mandataires authentifiés (pas d'accès anonyme) et via HTTPS.

Pour limiter les risques de sécurité, nous vous recommandons de respecter les points suivants :

- Étendez le rôle d'exécution Lambda au plus petit ensemble d'autorisations possible.
- Dans la mesure du possible, veillez à ce que votre fonction Lambda accède à Amazon S3 via l'URL pré-signée fournie.

Configuration des stratégies IAM

Les points d'accès S3 prennent en charge les politiques de ressources AWS Identity and Access Management (IAM) qui vous permettent de contrôler l'utilisation du point d'accès par ressource,

par utilisateur ou selon d'autres conditions. Pour plus d'informations, consultez Configuration des politiques IAM pour les points d'accès Object Lambda.

Comportement de chiffrement

Étant donné que les points d'accès Object Lambda utilisent à la fois Amazon S3 et qu' AWS Lambda il existe des différences de comportement de chiffrement. Pour plus d'informations sur le comportement de chiffrement S3 par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

- Lorsque vous utilisez le chiffrement côté serveur S3 avec des points d'accès Object Lambda, l'objet est déchiffré avant d'être envoyé à Lambda. Une fois l'objet envoyé à Lambda, il est traité sous sa forme déchiffrée (dans le cas d'une requête `GET` ou `HEAD`).
- Pour empêcher la journalisation de la clé de chiffrement, S3 rejette les requêtes `GET` et `HEAD` pour les objets chiffrés côté serveur avec des clés fournies par le client (SSE-C). Toutefois, la fonction Lambda peut encore récupérer ces objets si elle a accès à la clé fournie par le client.
- Lorsque vous utilisez le chiffrement côté client S3 avec des points d'accès Object Lambda, veillez à ce que Lambda ait accès à la clé de chiffrement pour pouvoir déchiffrer et rechiffrer l'objet.

Sécurité des points d'accès

S3 Object Lambda utilise deux points d'accès, un point d'accès Object Lambda et un point d'accès S3 standard, appelé point d'accès de prise en charge. Lorsque vous effectuez une demande auprès d'un point d'accès Object Lambda, S3 appelle Lambda en votre nom ou délègue la demande au point d'accès de prise en charge, en fonction de la configuration S3 Object Lambda. Lorsque Lambda est appelé pour une demande, S3 génère une URL pré-signée sur votre objet en votre nom via le point d'accès de prise en charge. Votre fonction Lambda reçoit cette URL en entrée quand la fonction est appelée.

Vous pouvez définir votre fonction Lambda pour utiliser cette URL pré-signée pour récupérer l'objet d'origine, au lieu d'appeler S3 directement. Ce modèle vous permet d'appliquer de meilleures limites de sécurité à vos objets. Vous pouvez limiter l'accès direct des objets à un ensemble limité de rôles ou d'utilisateurs IAM par le biais de compartiments S3 ou de points d'accès S3. Cette approche protège également vos fonctions Lambda face au problème de l'adjoint confus, dans le cadre duquel une fonction mal configurée, dotée d'autorisations différentes de l'appelant, pourrait autoriser ou refuser l'accès aux objets alors qu'elle ne le devrait pas.

Accès public au point d'accès Object Lambda

S3 Object Lambda n'autorise pas un accès anonyme ou public, car Amazon S3 doit autoriser votre identité pour effectuer une demande S3 Object Lambda quelconque. Lorsque vous appelez des demandes via un point d'accès Object Lambda, vous devez avoir l'autorisation `lambda:InvokeFunction` pour la fonction Lambda configurée. De même, lorsque vous appelez d'autres opérations d'API via un point d'accès Object Lambda, vous devez disposer des autorisations `s3:*` requises.

Sans ces autorisations, les demandes d'appel de Lambda et de délégation à S3 échouent avec des erreurs HTTP 403 (Interdit). Tous les accès doivent être effectués par des mandants authentifiés. Si vous avez besoin d'un accès public, vous pouvez utiliser Lambda@Edge comme alternative. Pour plus d'informations, consultez la section Personnalisation en périphérie avec Lambda @Edge dans le manuel CloudFront Amazon Developer Guide.

Adresses IP des points d'accès Object Lambda

Les sous-réseaux `describe-managed-prefix-lists` prennent en charge les points de terminaison de cloud privé virtuel (VPC) de la passerelle et sont liés à la table de routage des points de terminaison du VPC. Comme le point d’accès Object Lambda ne prend pas en charge le VPC de passerelle, ses plages IP sont manquantes. Les plages manquantes appartiennent à Amazon S3, mais ne sont pas prises en charge par les points de terminaison de VPC de la passerelle. Pour plus d'informations sur`describe-managed-prefix-lists`, consultez DescribeManagedPrefixListsla référence de l'API Amazon EC2 et les plages d'adresses AWS IP dans le. Références générales AWS

Configuration des politiques IAM pour les points d'accès Object Lambda

Les points d'accès Amazon S3 prennent en charge les politiques de ressources AWS Identity and Access Management (IAM) que vous pouvez utiliser pour contrôler l'utilisation du point d'accès par ressource, par utilisateur ou selon d'autres conditions. Vous pouvez contrôler l'accès par le biais d'une politique de ressources facultative sur votre point d'accès Object Lambda ou d'une politique de ressources sur le point d'accès de prise en charge. Pour step-by-step des exemples, voir Didacticiel : Transformation de données pour votre application avec S3 Object Lambda etDidacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend.

Les quatre ressources suivantes doivent disposer d'autorisations pour utiliser les points d'accès Object Lambda :

- L'identité IAM, telle que l'utilisateur ou le rôle. Pour obtenir plus d'informations sur les identités IAM et sur les bonnes pratiques, consultez la section IAM Identities (users, user groups, and roles) [Identités IAM (utilisateurs, groupes d'utilisateurs et rôles)] dans le Guide de l'utilisateur IAM.
- Le compartiment et le point d'accès standard associé. Lorsque vous utilisez des points d'accès Object Lambda, ce point d'accès standard est appelé point d'accès de prise en charge.
- Le point d'accès Object Lambda.
- La AWS Lambda fonction.

> **Important**
>
> Avant d'enregistrer votre politique, assurez-vous de résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions de AWS Identity and Access Management Access Analyzer. IAM Access Analyzer exécute des vérifications de politiques pour valider votre politique par rapport à la grammaire de politique et aux bonnes pratiques IAM. Ces vérifications génèrent des résultats et fournissent des recommandations exploitables pour vous aider à créer des stratégies fonctionnelles et conformes aux bonnes pratiques en matière de sécurité.
> Pour en savoir plus sur la validation des politiques à l'aide d'IAM Access Analyzer, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM. Pour afficher la liste des avertissements, erreurs et suggestions renvoyés par IAM Access Analyzer, consultez la Référence de vérification de stratégie IAM Access Analyzer.

Les exemples de politique suivants supposent que vous disposez des ressources suivantes :

- Un compartiment Amazon S3 avec l'Amazon Resource Name (ARN) suivant :

  `arn:aws:s3:::DOC-EXAMPLE-BUCKET1`

- Un point d'accès standard Simple Storage Service (Amazon S3) sur ce compartiment avec l'ARN suivant :

  `arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`

- Un point d'accès Object Lambda avec l'ARN suivant :

  `arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap`

- Une AWS Lambda fonction avec l'ARN suivant :

  `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

> **Note**
>
> Si vous utilisez une fonction Lambda depuis votre compte, vous devez inclure la version de la fonction spécifique dans votre déclaration de politique. Dans l'exemple d'ARN suivant, la version est indiquée par *1* :
> `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1`
> Lambda ne prend pas en charge l'ajout de politiques IAM à la version. $LATEST Pour plus d'informations sur les versions des fonctions Lambda, consultez Versions de fonctions Lambda dans le Guide du développeur AWS Lambda .

Example – Politique de compartiment qui délègue le contrôle d'accès aux points d'accès standard

L'exemple de politique de compartiment S3 suivant délègue le contrôle d'accès pour un compartiment aux points d'accès standard du compartiment. Cette politique permet un accès complet à tous les points d'accès appartenant au compte du propriétaire du compartiment. Ainsi, tous les accès à ce compartiment sont contrôlés par les politiques attachées à ses points d'accès. Les utilisateurs peuvent lire à partir du compartiment uniquement via un point d'accès, ce qui signifie que les opérations peuvent uniquement être appelées via des points d'accès. Pour plus d'informations, consultez Délégation du contrôle d'accès aux points d'accès.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS":"account-ARN"},
        "Action" : "*",
        "Resource" : [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"
        ],
        "Condition": {
```

```
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Example — Politique IAM qui accorde à un utilisateur les autorisations nécessaires pour utiliser un point d'accès Object Lambda

La politique IAM suivante accorde à un utilisateur des autorisations sur la fonction Lambda, le point d'accès standard et le point d'accès Object Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaInvocation",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyObjectLambdaFunction:1",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowStandardAccessPointAccess",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
```

```
          ]
        }
      }
    },
    {
      "Sid": "AllowObjectLambdaAccess",
      "Action": [
        "s3-object-lambda:Get*",
        "s3-object-lambda:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap"
    }
  ]
}
```

Activation des autorisations pour les rôles d'exécution Lambda

Lorsque des `GET` demandes sont adressées à un point d'accès Lambda d'objets, votre fonction Lambda a besoin d'une autorisation pour envoyer des données au point d'accès Lambda d'objets S3. Cette autorisation est fournie en activant l'autorisation `s3-object-lambda:WriteGetObjectResponse` sur le rôle d'exécution de votre fonction Lambda. Vous pouvez créer un rôle d'exécution ou mettre à jour un rôle existant.

> **Note**
>
> Votre fonction a besoin de l'autorisation `s3-object-lambda:WriteGetObjectResponse` uniquement si vous réalisez une requête `GET`.

Pour créer un rôle d'exécution dans la console IAM

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Rôles.
3. Sélectionnez Create role (Créer un rôle).
4. Sous Cas d'utilisation courants, choisissez Lambda.
5. Choisissez Suivant.

6. Sur la page Ajouter des autorisations, recherchez la stratégie AWS gérée `AmazonS3ObjectLambdaExecutionRolePolicy`, puis cochez la case à côté du nom de la stratégie.

   Cette politique doit contenir l'action `s3-object-lambda:WriteGetObjectResponse`.
7. Choisissez Suivant.
8. Sur la page Name, review, and create (Nommer, vérifier et créer), pour Role name (Nom du rôle), saisissez **`s3-object-lambda-role`**.
9. (Facultatif) Ajoutez une description et des balises pour ce rôle.
10. Sélectionnez Créer un rôle.
11. Appliquez l'objet récemment créé **`s3-object-lambda-role`** comme rôle d'exécution de votre fonction Lambda. Cela peut être fait pendant ou après la création de la fonction Lambda dans la console Lambda.

Pour plus d'informations sur les rôles d'exécution, consultez Rôle d'exécution Lambda dans le Guide du développeur AWS Lambda .

Utilisation des clés de contexte avec des points d'accès Object Lambda

S3 Object Lambda évaluera les clés de contexte telles que `s3-object-lambda:TlsVersion` ou `s3-object-lambda:AuthType` qui sont liées à la connexion ou à la signature de la demande. Toutes les autres clés de contexte, telles que `s3:prefix`, sont évaluées par Simple Storage Service (Amazon S3).

Prise en charge du CORS du point d'accès Object Lambda

Quand S3 Object Lambda reçoit une demande d'un navigateur ou que la demande inclut un en-tête `Origin`, S3 Object Lambda ajoute toujours un champ d'en-tête `“AllowedOrigins":"*"`.

Pour plus d’informations, consultez Utilisation du partage des ressources entre origines multiples (CORS).

Écriture de fonctions Lambda pour les points d'accès S3 Object Lambda

Cette section explique comment écrire des AWS Lambda fonctions à utiliser avec les points d'accès Amazon S3 Object Lambda.

Pour en savoir plus sur end-to-end les procédures complètes relatives à certaines tâches S3 Object Lambda, consultez les rubriques suivantes :

- Didacticiel : Transformation de données pour votre application avec S3 Object Lambda
- Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend
- Tutoriel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération

Rubriques

- Utilisation de requêtes GetObject dans Lambda
- Utilisation de requêtes HeadObject dans Lambda
- Utilisation de requêtes ListObjects dans Lambda
- Utilisation de requêtes ListObjectsV2 dans Lambda
- Format et utilisation du contexte d'événement
- Utilisation des en-têtes Range et partNumber

Utilisation de requêtes **`GetObject`** dans Lambda

Cette section suppose que votre point d'accès Object Lambda est configuré pour appeler la fonction Lambda pour `GetObject`. S3 Object Lambda inclut l'opération d'API Amazon S3, `WriteGetObjectResponse`, qui permet à la fonction Lambda de fournir des données personnalisées et des en-têtes de réponse à l'appelant `GetObject`.

`WriteGetObjectResponse` vous offre un contrôle étendu sur le code de statut, les en-têtes de réponse et le corps de réponse, en fonction de vos besoins de traitement. Vous pouvez utiliser `WriteGetObjectResponse` pour répondre avec l'ensemble de l'objet transformé, des parties de l'objet transformé ou d'autres réponses basées sur le contexte de votre application. La section suivante présente des exemples uniques d'utilisation de l'opération d'API `WriteGetObjectResponse`.

- Exemple 1 : répondre avec un code de statut HTTP 403 (Interdit)
- Exemple 2 : répondre avec une image transformée
- Exemple 3 : diffuser du contenu compressé

Exemple 1 : répondre avec un code de statut HTTP 403 (Interdit)

Vous pouvez utiliser `WriteGetObjectResponse` pour répondre avec le code d'état HTTP 403 (Interdit) en fonction du contenu de l'objet.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                    .withRequestRoute(event.outputRoute())
                    .withRequestToken(event.outputToken())
                    .withStatusCode(403)
                    .withContentLength(0L).withInputStream(new
 ByteArrayInputStream(new byte[0]))
                    .withErrorCode("MissingRequiredToken")
                    .withErrorMessage("The required token was not present in the
 request."));
            return;
        }

        // Prepare the presigned URL for use and make the request to S3.
        HttpClient httpClient = HttpClient.newBuilder().build();
```

```
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // Stream the original bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(presignedResponse.body()));
    }
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    user_request_headers = event["userRequest"]["headers"]

    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    is_token_present = "SuperSecretToken" in user_request_headers

    if is_token_present:
        # If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
```

```
        response = requests.get(s3_url)
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
Body=response.content)
    else:
        # If the token is not present, we send an error back to the user.
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
StatusCode=403,
        ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not
secret enough.")

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    // The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
    const { userRequest, getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    const isTokenPresent = Object
        .keys(userRequest.headers)
        .includes("SuperSecretToken");

    if (!isTokenPresent) {
        // If the token is not present, we send an error back to the user. The
'await' in front of the request
        // indicates that we want to wait for this request to finish sending before
 moving on.
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
```

```
            RequestToken: outputToken,
            StatusCode: 403,
            ErrorCode: "NoSuperSecretTokenFound",
            ErrorMessage: "The request was not secret enough.",
        }).promise();
    } else {
        // If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
        // Again, note the presence of 'await'.
        const presignedResponse = await axios.get(inputS3Url);
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: presignedResponse.data,
        }).promise();
    }

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Exemple 2 : répondre avec une image transformée

Lorsque vous effectuez une transformation d'image, vous pouvez constater que vous avez besoin de tous les octets de l'objet source avant de pouvoir commencer à les traiter. Dans ce cas, votre demande `WriteGetObjectResponse` renvoie l'objet entier à l'application requérante en un seul appel.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
```

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {

    private static final int HEIGHT = 250;
    private static final int WIDTH = 250;

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Prepare the presigned URL for use and make the request to S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // The entire image is loaded into memory here so that we can resize it.
        // Once the resizing is completed, we write the bytes into the body
        // of the WriteGetObjectResponse request.
        var originalImage = ImageIO.read(presignedResponse.body());
        var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
 Image.SCALE_DEFAULT);
        var resizedImage = new BufferedImage(WIDTH, HEIGHT,
 BufferedImage.TYPE_INT_RGB);
        resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
 null);

        var baos = new ByteArrayOutputStream();
        ImageIO.write(resizedImage, "png", baos);

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
    }
}
```

Python

```
import boto3
import requests
import io
from PIL import Image

def handler(event, context):
    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
    In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
 RequestToken=token)

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const sharp = require('sharp');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    // 'inputS3Url'.
    const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });

    // Resize the image.
    const resized = await sharp(data)
        .resize({ width: 256, height: 256 })
        .toBuffer();

    // Send the resized image back to the client.
    await s3.writeGetObjectResponse({
        RequestRoute: outputRoute,
        RequestToken: outputToken,
        Body: resized,
    }).promise();

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Exemple 3 : diffuser du contenu compressé

Lorsque vous compressez des objets, les données compressées sont produites de manière incrémentielle. Par conséquent, vous pouvez utiliser votre demande `WriteGetObjectResponse`

pour renvoyer les données compressées dès qu'elles sont prêtes. Comme le montre cet exemple, vous n'avez pas besoin de connaître la longueur de la transformation terminée.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // Consume the incoming response body from the presigned request,
        // apply our transformation on that data, and emit the transformed bytes
        // into the body of the WriteGetObjectResponse request as soon as they're
 ready.
        // This example compresses the data from S3, but any processing pertinent
        // to your application can be performed here.
        var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(bodyStream));
```

```
    }

}
```

Python

```
import boto3
import requests
import zlib
from botocore.config import Config


"""
A helper class to work with content iterators. Takes an interator and compresses the
 bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__()
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()


def handler(event, context):
    """
    Setting the 'payload_signing_enabled' property to False allows us to send a
 streamed response back to the client.
```

```
   in this scenario, a streamed response means that the bytes are not buffered into
memory as we're compressing them,
   but instead are sent straight to the user.
   """
   my_config = Config(
       region_name='eu-west-1',
       signature_version='s3v4',
       s3={
           "payload_signing_enabled": False
       }
   )
   s3 = boto3.client('s3', config=my_config)

   """
   Retrieve the operation context object from the event. This object indicates
where the WriteGetObjectResponse request
   should be delivered and has a presigned URL in 'inputS3Url' where we can
download the requested object from.
   The 'userRequest' object has information related to the user who made this
'GetObject' request to S3 Object Lambda.
   """
   get_context = event["getObjectContext"]
   route = get_context["outputRoute"]
   token = get_context["outputToken"]
   s3_url = get_context["inputS3Url"]

   # Compress the 'get' request stream.
   with requests.get(s3_url, stream=True) as r:
       compressed = Compress(r.iter_content())

       # Send the stream back to the client.
       s3.write_get_object_response(Body=compressed, RequestRoute=route,
RequestToken=token, ContentType="text/plain",
                                    ContentEncoding="gzip")

   # Gracefully exit the Lambda function.
   return {'status_code': 200}
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const zlib = require('zlib');
```

```
exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Download the object from S3 and process it as a stream, because it might be a
 huge object and we don't want to
    // buffer it in memory. Note the use of 'await' because we want to wait for
 'writeGetObjectResponse' to finish
    // before we can exit the Lambda function.
    await axios({
        method: 'GET',
        url: inputS3Url,
        responseType: 'stream',
    }).then(
        // Gzip the stream.
        response => response.data.pipe(zlib.createGzip())
    ).then(
        // Finally send the gzip-ed stream back to the client.
        stream => s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: stream,
            ContentType: "text/plain",
            ContentEncoding: "gzip",
        }).promise()
    );

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

> **Note**
>
> Bien que S3 Object Lambda laisse jusqu'à 60 secondes pour envoyer une réponse complète à l'appelant via la demande `WriteGetObjectResponse`, le temps réel disponible peut être

inférieur. Par exemple, le délai d'expiration de votre fonction Lambda peut être inférieur à 60 secondes. Dans d'autres cas, l'appelant peut avoir des délais d'attente plus stricts.

Pour que l'appelant d'origine reçoive une réponse autre que le code de statut HTTP 500 (Erreur de serveur interne), l'appel `WriteGetObjectResponse` doit être terminé. Si la fonction Lambda renvoie une réponse, avec une exception ou autrement, avant que l'opération d'API `WriteGetObjectResponse` soit appelée, l'appelant d'origine reçoit une réponse 500 (Erreur de serveur interne). Les exceptions lancées pendant le temps nécessaire pour compléter la réponse entraînent des réponses tronquées à l'appelant. Si la fonction Lambda reçoit une réponse avec code de statut HTTP 200 (OK) de l'appel d'API `WriteGetObjectResponse`, l'appelant d'origine a envoyé la demande complète. La réponse de la fonction Lambda, qu'une exception soit déclenchée ou non, est ignorée par S3 Object Lambda.

Au moment d'appeler l'opération d'API `WriteGetObjectResponse`, Amazon S3 exige la route et le jeton de demande du contexte d'événement. Pour plus d'informations, consultez Format et utilisation du contexte d'événement.

Les paramètres de route et de jeton de demande sont nécessaires pour connecter la réponse `WriteGetObjectResult` à l'appelant d'origine. Bien qu'il soit toujours approprié de réessayer les réponses 500 (Erreur de serveur interne), car le jeton de demande est un jeton à usage unique, les tentatives ultérieures de l'utiliser peuvent entraîner des réponses avec code de statut HTTP 400 (Demande incorrecte). Bien que l'appel à `WriteGetObjectResponse` avec la route et les jetons de demande n'ait pas besoin d'être réalisé à partir de la fonction Lambda appelée, il doit être réalisé par une identité figurant dans le même compte. L'appel doit également être terminé avant que la fonction Lambda ne termine l'exécution.

Utilisation de requêtes **`HeadObject`** dans Lambda

Cette section suppose que votre point d'accès Object Lambda est configuré pour appeler la fonction Lambda pour `HeadObject`. Lambda recevra une charge utile JSON contenant une clé appelée `headObjectContext`. Dans le contexte, il existe une propriété unique appelée `inputS3Url`, qui est une URL pré-signée pour le point d'accès de prise en charge pour `HeadObject`.

L'URL pré-signée inclura les propriétés suivantes si elles sont spécifiées :

- `versionId` (dans les paramètres de la requête)
- `requestPayer` (dans l'en-tête `x-amz-request-payer`)
- `expectedBucketOwner` (dans l'en-tête `x-amz-expected-bucket-owner`)

Les autres propriétés ne seront pas pré-signées et ne seront donc pas incluses. Les options non signées envoyées sous forme d'en-têtes peuvent être ajoutées manuellement à la demande lors de l'appel de l'URL pré-signée qui se trouve dans les en-têtes `userRequest`. Les options de chiffrement côté serveur ne sont pas prises en charge pour `HeadObject`.

Pour les paramètres d'URI de la syntaxe de la demande, consultez `HeadObject` dans la Référence des API Amazon Simple Storage Service.

L'exemple suivant illustre une charge utile d'entrée JSON de Lambda pour `HeadObject`.

```
{
  "xAmzRequestId": "requestId",
  "**headObjectContext**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>"
  },
  "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
  },
  "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
   },
   "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
             "attributes": {
             "mfaAuthenticated": "false",
             "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
```

```
        },
        "sessionIssuer": {
             "type": "Role",
             "principalId": "principalId",
             "arn": "arn:aws:iam::111122223333:role/Admin",
             "accountId": "111122223333",
             "userName": "Admin"
             }
        }
    },
  "protocolVersion": "1.00"
}
```

Votre fonction Lambda doit renvoyer un objet JSON contenant les en-têtes et les valeurs qui seront renvoyés pour l'appel `HeadObject`.

L'exemple suivant illustre la structure du code JSON de réponse de Lambda pour `HeadObject`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "headers": {
        "Accept-Ranges": <string>,
        "x-amz-archive-status": <string>,
        "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
        "Cache-Control": <string>,
        "Content-Disposition": <string>,
        "Content-Encoding": <string>,
        "Content-Language": <string>,
        "Content-Length": <number>, // Required
        "Content-Type": <string>,
        "x-amz-delete-marker": <boolean>,
        "ETag": <string>,
        "Expires": <string>,
        "x-amz-expiration": <string>,
        "Last-Modified": <string>,
        "x-amz-missing-meta": <number>,
        "x-amz-object-lock-mode": <string>,
        "x-amz-object-lock-legal-hold": <string>,
        "x-amz-object-lock-retain-until-date": <string>,
        "x-amz-mp-parts-count": <number>,
        "x-amz-replication-status": <string>,
```

```
        "x-amz-request-charged": <string>,
        "x-amz-restore": <string>,
        "x-amz-server-side-encryption": <string>,
        "x-amz-server-side-encryption-customer-algorithm": <string>,
        "x-amz-server-side-encryption-aws-kms-key-id": <string>,
        "x-amz-server-side-encryption-customer-key-MD5": <string>,
        "x-amz-storage-class": <string>,
        "x-amz-tagging-count": <number>,
        "x-amz-version-id": <string>,
        <x-amz-meta-headers>: <string>, // user-defined metadata
        "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
 it will need the x-amz-meta prefix
        "x-amz-meta-meta2": <string>
        ...
    };
}
```

L'exemple suivant montre comment utiliser l'URL pré-signée pour renseigner votre réponse en modifiant les valeurs d'en-tête selon les besoins avant de renvoyer le code JSON.

Python

```
import requests

def lambda_handler(event, context):
    print(event)

    # Extract the presigned URL from the input.
    s3_url = event["headObjectContext"]["inputS3Url"]

    # Get the head of the object from S3.
    response = requests.head(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        return {
            "statusCode": response.status_code,
            "errorCode": "RequestFailure",
            "errorMessage": "Request to S3 failed"
    }

    # Store the headers in a dictionary.
    response_headers = dict(response.headers)
```

```
    # This obscures Content-Type in a transformation, it is optional to add
    response_headers["Content-Type"] = ""

    # Return the headers to S3 Object Lambda.
    return {
        "statusCode": response.status_code,
        "headers": response_headers
        }
```

Utilisation de requêtes **`ListObjects`** dans Lambda

Cette section suppose que votre point d'accès Object Lambda est configuré pour appeler la fonction Lambda pour `ListObjects`. Lambda recevra la charge utile JSON avec un nouvel objet nommé `listObjectsContext`. `listObjectsContext` contient une propriété unique, `inputS3Url`, qui est une URL pré-signée pour le point d'accès de prise en charge pour `ListObjects`.

Contrairement à `GetObject` et `HeadObject`, l'URL pré-signée inclura les propriétés suivantes si elles sont spécifiées :

- Tous les paramètres de la requête
- `requestPayer` (dans l'en-tête `x-amz-request-payer`)
- `expectedBucketOwner` (dans l'en-tête `x-amz-expected-bucket-owner`)

Pour les paramètres d'URI de la syntaxe de la demande, consultez `ListObjects` dans la Référence des API Amazon Simple Storage Service.

> **Important**
>
> Nous vous recommandons d'utiliser la version la plus récente, ListObjectsV2, lorsque vous développez des applications. Pour assurer la compatibilité descendante, Amazon S3 continue de prendre en charge `ListObjects`.

L'exemple suivant illustre la charge utile d'entrée JSON de Lambda pour `ListObjects`.

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsContext**": {
```

```
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?X-Amz-Security-Token=<snip>",
     },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
  "protocolVersion": "1.00"
}
```

Votre fonction Lambda devrait renvoyer un objet JSON contenant le code de statut, le résultat XML de liste ou les informations d'erreur qui seront renvoyées de S3 Object Lambda.

S3 Object Lambda ne traite ni ne valide `listResultXml`, mais les transmet à l'appelant `ListObjects`. Pour `listBucketResult`, S3 Object Lambda s'attend à ce que certaines propriétés soient d'un type spécifique et lèvera des exceptions s'il ne peut pas les analyser. `listResultXml` et `listBucketResult` ne peuvent pas être fournies en même temps.

L'exemple suivant montre comment utiliser l'URL pré-signée pour appeler Amazon S3 et comment utiliser le résultat pour renseigner une réponse ainsi que vérifier les erreurs.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsContext"]["inputS3Url"]


    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)
```

```
    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

L'exemple suivant illustre la structure du code JSON de réponse de Lambda pour `ListObjects`.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
```

```
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL

  "listBucketResult": {  // listBucketResult can be provided instead of listResultXml,
 however they can not both be provided in the JSON response
        "name": <string>,  // Required for 'listBucketResult'
        "prefix": <string>,
        "marker": <string>,
        "nextMarker": <string>,
        "maxKeys": <int>,   // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>,  // Required for 'listBucketResult'
        "contents": [  {
            "key": <string>,  // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>,   // CRC32,  CRC32C,  SHA1,  SHA256
            "size": <int>,   // Required for 'content'
            "owner": {
                "displayName": <string>,  // Required for 'owner'
                "id": <string>,  // Required for 'owner'
            },
            "storageClass": <string>
            },
        ...
        ],
        "commonPrefixes": [  {
            "prefix": <string>   // Required for 'commonPrefix'
        },
        ...
        ],
    }
}
```

Utilisation de requêtes **`ListObjectsV2`** dans Lambda

Cette section suppose que votre point d'accès Object Lambda est configuré pour appeler la fonction Lambda pour `ListObjectsV2`. Lambda recevra la charge utile JSON avec un nouvel objet nommé `listObjectsV2Context`. `listObjectsV2Context` contient une propriété unique, `inputS3Url`, qui est une URL pré-signée pour le point d'accès de prise en charge pour `ListObjectsV2`.

Contrairement à `GetObject` et `HeadObject`, l'URL pré-signée inclura les propriétés suivantes, si elles sont spécifiées :

- Tous les paramètres de la requête
- `requestPayer` (dans l'en-tête `x-amz-request-payer`)
- `expectedBucketOwner` (dans l'en-tête `x-amz-expected-bucket-owner`)

Pour les paramètres d'URI de la syntaxe de la demande, consultez `ListObjectsV2` dans la Référence des API Amazon Simple Storage Service.

L'exemple suivant illustre la charge utile d'entrée JSON de Lambda pour `ListObjectsV2`.

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsV2Context**": {
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?list-type=2&X-Amz-Security-Token=<snip>",
     },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
```

```
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
  "protocolVersion": "1.00"
}
```

Votre fonction Lambda devrait renvoyer un objet JSON contenant le code de statut, le résultat XML de liste ou les informations d'erreur qui seront renvoyées de S3 Object Lambda.

S3 Object Lambda ne traite ni ne valide `listResultXml`, mais les transmet à l'appelant `ListObjectsV2`. Pour `listBucketResult`, S3 Object Lambda s'attend à ce que certaines propriétés soient d'un type spécifique et lèvera des exceptions s'il ne peut pas les analyser. `listResultXml` et `listBucketResult` ne peuvent pas être fournies en même temps.

L'exemple suivant montre comment utiliser l'URL pré-signée pour appeler Amazon S3 et comment utiliser le résultat pour renseigner une réponse ainsi que vérifier les erreurs.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsV2Context"]["inputS3Url"]


    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
```

```
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
```

```
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

L'exemple suivant illustre la structure du code JSON de réponse de Lambda pour `ListObjectsV2`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL

    "listBucketResult": {  // listBucketResult can be provided instead of
 listResultXml, however they can not both be provided in the JSON response
        "name": <string>, // Required for 'listBucketResult'
        "prefix": <string>,
        "startAfter": <string>,
        "continuationToken": <string>,
        "nextContinuationToken": <string>,
        "keyCount": <int>, // Required for 'listBucketResult'
        "maxKeys": <int>, // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>, // Required for 'listBucketResult'
        "contents": [ {
            "key": <string>, // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
            "size": <int>, // Required for 'content'
            "owner": {
                "displayName": <string>, // Required for 'owner'
                "id": <string>, // Required for 'owner'
            },
            "storageClass": <string>
            },
            ...
        ],
```

```
        "commonPrefixes": [ {
            "prefix": <string> // Required for 'commonPrefix'
            },
        ...
        ],
    }
}
```

Format et utilisation du contexte d'événement

Amazon S3 Object Lambda fournit le contexte de la demande effectuée lors de l'événement transmis à votre AWS Lambda fonction. Voici un exemple de demande. Les descriptions des champs sont incluses après l'exemple.

```
{
    "xAmzRequestId": "requestId",
    "getObjectContext": {
        "inputS3Url": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>",
        "outputRoute": "io-use1-001",
        "outputToken": "OutputToken"
    },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
```

```
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
    "protocolVersion": "1.00"
}
```

Les champs suivants sont inclus dans la demande :

- `xAmzRequestId` – ID de demande Amazon S3 pour cette demande. Nous vous recommandons de consigner cette valeur pour faciliter le débogage.
- `getObjectContext` – Détails d'entrée et de sortie pour les connexions à Amazon S3 et S3 Object Lambda.
  - `inputS3Url` – URL pré-signée qui peut être utilisée pour récupérer l'objet d'origine depuis Amazon S3. L'URL est signée avec l'identité de l'appelant d'origine et les autorisations de cet utilisateur s'appliquent quand l'URL est utilisée. S'il y a des en-têtes signés dans cette URL, la fonction Lambda doit les inclure dans l'appel à Amazon S3, à l'exception de l'en-tête `Host`.
  - `outputRoute` – Un jeton de routage qui est ajouté à l'URL S3 Object Lambda lorsque la fonction Lambda appelle `WriteGetObjectResponse`.
  - `outputToken` – Jeton opaque utilisé par S3 Object Lambda pour mettre en correspondance l'appel `WriteGetObjectResponse` avec l'appelant d'origine.
- `configuration` : informations de configuration sur le point d'accès Object Lambda.
  - `accessPointArn` : Amazon Resource Name (ARN) du point d'accès Object Lambda qui a reçu cette demande.
  - `supportingAccessPointArn` : ARN du point d'accès de prise en charge spécifié dans la configuration du point d'accès Object Lambda.

  - `payload` : données personnalisées qui sont appliquées à la configuration du point d'accès Object Lambda. S3 Object Lambda considère ces données comme une chaîne opaque. Dès lors, elles devront peut-être être décodées avant utilisation.
- `userRequest` – Informations sur l'appel d'origine à S3 Object Lambda.
  - `url` – URL décodée de la demande telle qu'elle a été reçue par S3 Object Lambda, à l'exclusion des paramètres de requête liés à l'autorisation.
  - `headers` – Carte de chaîne aux chaînes contenant les en-têtes HTTP et leurs valeurs de l'appel d'origine, à l'exclusion de tous les en-têtes liés à l'autorisation. Si le même en-tête apparaît plusieurs fois, les valeurs de chaque instance du même en-tête sont combinées dans une liste délimitée par des virgules. Le cas des en-têtes d'origine est conservé dans cette carte.
- `userIdentity` – Détails sur l'identité qui a adressé l'appel à S3 Object Lambda. Pour plus d'informations, veuillez consulter Consignation d'événements de données pour les journaux d'activité dans le Guide de l'utilisateur AWS CloudTrail .
  - `type` – Type d'identité.
  - `accountId`— Compte AWS À laquelle appartient l'identité.
  - `userName` – Nom descriptif de l'identité qui a réalisé l'appel.
  - `principalId` – Identifiant unique de l'entité qui a effectué l'appel.
  - `arn` – ARN du principal qui a effectué l'appel. La dernière partie de l'ARN contient l'utilisateur ou le rôle qui a réalisé l'appel.
  - `sessionContext` – Si la demande a été effectuée avec les autorisations de sécurité temporaires, cet élément fournit des informations sur la session qui a été créée pour ces autorisations.
  - `invokedBy`— Le nom de la personne à Service AWS l'origine de la demande, par exemple Amazon EC2 Auto Scaling ou. AWS Elastic Beanstalk
  - `sessionIssuer` – Si la demande a été effectuée avec des informations d'identification de sécurité temporaires, cet élément fournit des informations sur la façon dont les autorisations ont été obtenues.
- `protocolVersion` – ID de version du contexte fourni. Le format de ce champ est `{Major Version}.{Minor Version}`. Les numéros de version mineure sont toujours des nombres à deux chiffres. Toute suppression ou modification de la sémantique d'un champ nécessite un changement de version majeur et une participation active. Simple Storage Service (Amazon S3) peut ajouter de nouveaux champs à tout moment, auquel cas vous risquez de rencontrer un changement de version mineure. En raison de la nature des déploiements logiciels, il est possible que plusieurs versions mineures soient utilisées simultanément.

Utilisation des en-têtes Range et partNumber

Lorsque vous travaillez avec des objets volumineux dans Amazon S3 Object Lambda, vous pouvez utiliser l'en-tête HTTP `Range` pour télécharger une plage d'octets spécifiée à partir d'un objet. Pour extraire différentes plages d'octets à partir d'un même objet, utilisez des connexions simultanées à Amazon S3. Spécifiez également le paramètre `partNumber` (entier compris entre 1 et 10 000) pour exécuter une demande « par plage » pour la partie spécifiée de l'objet.

Étant donné qu'il existe plusieurs façons de gérer une demande qui inclut le paramètre `Range` ou `partNumber`, S3 Object Lambda n'applique pas ces paramètres à l'objet transformé. Votre AWS Lambda fonction doit plutôt implémenter cette fonctionnalité selon les besoins de votre application.

Pour utiliser les paramètres `Range` et `partNumber` avec S3 Object Lambda, procédez comme suit :

- Activez ces paramètres dans la configuration de vos points d'accès Object Lambda.
- Écrivez une fonction Lambda capable de gérer les demandes qui incluent ces paramètres.

Les étapes suivantes expliquent comment procéder.

Étape 1 : Configurer votre point d'accès Object Lambda

Par défaut, les points d'accès Object Lambda répondent avec une erreur de code de statut HTTP 501 (Non implémenté) à toute requête `GetObject` ou `HeadObject` contenant un paramètre `Range` ou `partNumber`, que ce soit dans les en-têtes ou les paramètres de requête.

Pour permettre à un point d'accès Object Lambda d'accepter ces demandes, vous devez inclure `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` ou `HeadObject-PartNumber` dans la section `AllowedFeatures` de la configuration de votre point d'accès Object Lambda. Pour plus d'informations sur la mise à jour de votre configuration de point d'accès Object Lambda, consultez Création de points d'accès Object Lambda.

Étape 2 : implémenter la gestion de **`Range`** ou **`partNumber`** dans votre fonction Lambda

Lorsque votre point d'accès Object Lambda appelle votre fonction Lambda avec une demande `GetObject` ou `HeadObject` par plage, le paramètre `Range` ou `partNumber` est inclus dans le contexte de l'événement. L'emplacement du paramètre dans le contexte d'événement dépend du paramètre utilisé et de la façon dont il a été inclus dans la demande d'origine au point d'accès Object Lambda, comme expliqué dans le tableau suivant.

Paramètre	Emplacement du contexte d'événement
`Range` (en-tête)	`userRequest.headers.Range`
`Range` (paramètre de la demande)	`userRequest.url` (paramètre `Range` de la demande)
`partNumber`	`userRequest.url` (paramètre `partNumber` de la demande)

> **Important**
>
> L'URL pré-signée fournie pour votre point d'accès Object Lambda ne contient pas le paramètre `Range` ou `partNumber` de la demande d'origine. Consultez les options suivantes pour savoir comment gérer ces paramètres dans votre AWS Lambda fonction.

Après avoir extrait la valeur `Range` ou `partNumber`, vous pouvez adopter l'une des approches suivantes en fonction des besoins de votre application :

A. Mapper le paramètre **`Range`** ou **`partNumber`** demandé à l'objet transformé (recommandé)

La méthode la plus fiable pour traiter les requêtes `Range` et `partNumber` consiste à effectuer les opérations suivantes :

- Récupérez l'objet complet à partir d'Amazon S3.
- Transformez l'objet.
- Appliquez le paramètre `Range` ou `partNumber` demandé à l'objet transformé.

Pour ce faire, utilisez l'URL pré-signée fournie afin de récupérer l'objet entier d'Amazon S3, puis traitez l'objet au besoin. Pour un exemple de fonction Lambda qui traite un `Range` paramètre de cette manière, consultez cet exemple dans le référentiel AWS Samples GitHub .

B. Mappez l'élément **`Range`** demandé sur l'URL pré-signée.

Dans certains cas, votre fonction Lambda peut mapper l'élément `Range` demandé directement sur l'URL pré-signée pour récupérer uniquement une partie de l'objet à partir d'Amazon S3. Cette approche n'est appropriée que si votre transformation répond aux deux critères suivants :

1. Votre fonction de transformation peut être appliquée à des plages d'objets partielles.

2. L'application du paramètre `Range` avant ou après la fonction de transformation aboutit au même objet transformé.

Par exemple, une fonction de transformation qui convertit tous les caractères d'un objet codé en ASCII en majuscules répond aux deux critères précédents. La transformation peut être appliquée à une partie d'un objet, et l'application du paramètre `Range` avant la transformation donne le même résultat que son application après la transformation.

En revanche, une fonction qui rétablit les caractères d'un objet codé en ASCII ne répond pas à ces critères. Cette fonction répond au critère 1, car elle peut être appliquée à des plages d'objets partielles. Cependant, elle ne répond pas au critère 2, car le paramètre `Range` appliqué avant la transformation ne donne pas le même résultat que lorsqu'il est appliqué après la transformation.

Prenons une demande d'application de la fonction aux trois premiers caractères d'un objet avec le contenu `abcdefg`. Si le paramètre `Range` est appliqué avant la transformation, il récupère uniquement `abc`, puis rétablit les données et renvoie `cba`. Mais si le paramètre est appliqué après la transformation, la fonction récupère l'objet entier, le rétablit, puis applique le paramètre `Range` et renvoie `gfe`. Étant donné que ces résultats sont différents, cette fonction ne doit pas appliquer le paramètre `Range` lors de la récupération de l'objet à partir d'Amazon S3. À la place, elle doit récupérer l'objet entier, effectuer la transformation, et seulement alors appliquer le paramètre `Range`.

> **Warning**
>
> Dans de nombreux cas, l'application du paramètre `Range` à l'URL pré-signée entraîne un comportement inattendu de la fonction Lambda ou du client demandeur. À moins que vous ne soyez sûr que votre application fonctionne correctement lorsque vous récupérez uniquement un objet partiel d'Amazon S3, nous vous recommandons de récupérer et de transformer des objets complets comme décrit précédemment dans l'approche A.

Si votre application répond aux critères décrits précédemment dans l'approche B, vous pouvez simplifier votre AWS Lambda fonction en récupérant uniquement la plage d'objets demandée, puis en exécutant votre transformation sur cette plage.

L'exemple de code Java suivant montre comment effectuer les opérations suivantes :

- Récupérer l'en-tête `Range` de la requête `GetObject`.

- Ajouter l'en-tête `Range` à l'URL pré-signée que Lambda peut utiliser pour récupérer la plage demandée à partir d'Amazon S3.

```
private HttpRequest.Builder applyRangeHeader(ObjectLambdaEvent event,
 HttpRequest.Builder presignedRequest) {
    var header = event.getUserRequest().getHeaders().entrySet().stream()
            .filter(e -> e.getKey().toLowerCase(Locale.ROOT).equals("range"))
            .findFirst();

    // Add check in the query string itself.
    header.ifPresent(entry -> presignedRequest.header(entry.getKey(),
 entry.getValue()));
    return presignedRequest;
}
```

Utilisation de AWS fonctions Lambda intégrées

AWS fournit des AWS Lambda fonctions prédéfinies que vous pouvez utiliser avec Amazon S3 Object Lambda pour détecter et supprimer les informations personnelles identifiables (PII) et décompresser les objets S3. Ces fonctions Lambda sont disponibles dans AWS Serverless Application Repository. Vous pouvez sélectionner ces fonctions via la AWS Management Console lorsque vous créez votre point d'accès Object Lambda.

Pour plus d'informations sur le déploiement d'applications sans serveur depuis le AWS Serverless Application Repository, consultez la section Déploiement d'applications dans le guide du AWS Serverless Application Repository développeur.

> **Note**
>
> Les exemples suivants peuvent être utilisés uniquement avec des requêtes `GetObject`.

Exemple 1 : contrôle d'accès aux données d'identification personnelle (PII)

Cette fonction Lambda utilise Amazon Comprehend, un service de traitement du langage naturel (NLP) qui a recours au machine learning pour identifier des informations et des relations dans un texte. Elle détecte automatiquement les données d'identification personnelle (PII), telles que les noms, adresses, dates, numéros de carte de crédit et numéros de sécurité sociale, dans les documents figurant dans votre compartiment Amazon S3. Si vous avez des documents dans votre

compartiment qui incluent des données PII, vous pouvez configurer la fonction PII Access Control pour détecter ces types d'entités PII et en bloquer l'accès aux utilisateurs non autorisés.

Pour commencer, déployez la fonction Lambda suivante dans votre compte et ajoutez l'Amazon Resource Name (ARN) de cette fonction dans la configuration de votre point d'accès Object Lambda.

Voici un exemple d'ARN pour cette fonction :

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/
ComprehendPiiAccessControlS3ObjectLambda
```

Vous pouvez ajouter ou consulter cette fonction sur AWS Management Console le AWS Serverless Application Repository lien suivant : ComprehendPiiAccessControlS3 ObjectLambda.

Pour afficher cette fonction GitHub, consultez Amazon Comprehend S3 Object Lambda.

Exemple 2 : rédaction de données d'identification personnelle (PII)

Cette fonction Lambda utilise Amazon Comprehend, un service de traitement du langage naturel (NLP) qui a recours au machine learning pour identifier des informations et des relations dans un texte. Elle supprime automatiquement les données d'identification personnelle (PII), telles que les noms, adresses, dates, numéros de carte de crédit et numéros de sécurité sociale, dans les documents figurant dans votre compartiment Amazon S3.

Si vous avez des documents dans votre compartiment qui incluent des informations telles que des numéros de carte de crédit ou des informations de compte bancaire, vous pouvez configurer la fonction S3 Object Lambda Rédaction de PII pour détecter les IPI, puis renvoyer une copie de ces documents dans lesquels les types d'entités PII sont expurgés.

Pour commencer, déployez la fonction Lambda suivante dans votre compte et ajoutez l'ARN de cette fonction dans la configuration de votre point d'accès Object Lambda.

Voici un exemple d'ARN pour cette fonction :

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/
ComprehendPiiRedactionS3ObjectLambda
```

Vous pouvez ajouter ou consulter cette fonction sur AWS Management Console le AWS Serverless Application Repository lien suivant : ComprehendPiiRedactionS3 ObjectLambda.

Pour afficher cette fonction GitHub, consultez Amazon Comprehend S3 Object Lambda.

Pour en savoir plus sur end-to-end les procédures complètes relatives à certaines tâches Lambda d'objets S3 dans la rédaction d'informations personnelles, consultez. Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend

Exemple 3 : Décompression

La fonction Lambda `S3ObjectLambdaDecompression` peut décompresser les objets stockés dans Amazon S3 dans l'un des six formats de fichiers compressés suivants : `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` ou `ZIP`.

Pour commencer, déployez la fonction Lambda suivante dans votre compte et ajoutez l'ARN de cette fonction dans la configuration de votre point d'accès Object Lambda.

Voici un exemple d'ARN pour cette fonction :

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

Vous pouvez ajouter ou consulter cette fonction sur AWS Management Console le AWS Serverless Application Repository lien suivant : S3 ObjectLambdaDecompression.

Pour afficher cette fonction GitHub, consultez la section Décompression Lambda d'objets S3.

Bonnes pratiques et directives pour S3 Object Lambda

Lorsque vous utilisez S3 Object Lambda, suivez ces bonnes pratiques et directives pour optimiser les opérations et les performances.

Rubriques

- Utilisation de S3 Object Lambda
- Services AWS utilisé en relation avec S3 Object Lambda
- En-têtes Range et partNumber
- Transformation de la date expiry-date
- Utilisation des AWS SDK AWS CLI et

Utilisation de S3 Object Lambda

S3 Object Lambda prend en charge uniquement le traitement des requêtes `GET`, `LIST` et `HEAD`. Toutes les autres requêtes n'invoquent AWS Lambda pas mais renvoient des réponses d'API

standard non transformées. Vous pouvez créer un maximum de 1 000 points d'accès Object Lambda Compte AWS par région. La AWS Lambda fonction que vous utilisez doit se trouver dans la même Compte AWS région que le point d'accès Object Lambda.

S3 Object Lambda accorde jusqu'à 60 secondes pour diffuser une réponse complète à son mandataire. Votre fonction est également soumise à des quotas AWS Lambda par défaut. Pour plus d'informations, consultez la section Quotas Lambda du Guide du développeur AWS Lambda .

Quand S3 Object Lambda appelle votre fonction Lambda spécifiée, il vous incombe de veiller à ce que toutes les données écrasées ou supprimées d'Amazon S3 par votre fonction ou application Lambda spécifiée sont volontaires et correctes.

Vous pouvez utiliser S3 Object Lambda uniquement pour effectuer des opérations sur des objets. Vous ne pouvez pas utiliser S3 Object Lambda pour effectuer d'autres opérations Amazon S3, comme modifier ou supprimer des compartiments. Pour obtenir la liste complète des opérations S3 qui prennent en charge les points d'accès, veuillez consulter Compatibilité des points d'accès avec les opérations S3.

Outre cette liste, les points d'accès Object Lambda ne prennent pas en charge les opérations d'API `POST Object`, `CopyObject` (en tant que source) ou `SelectObjectContent`.

Services AWS utilisé en relation avec S3 Object Lambda

S3 Object Lambda connecte Amazon S3 et, éventuellement AWS Lambda, les autres appareils Services AWS de votre choix pour fournir des objets adaptés aux applications demandeuses. Toutes les Services AWS applications utilisées avec S3 Object Lambda sont régies par leurs accords de niveau de service (SLA) respectifs. Par exemple, si l'un d'entre eux Service AWS ne respecte pas son engagement de service, vous pouvez bénéficier d'un crédit de service, comme indiqué dans le SLA du service.

En-têtes **`Range`** et **`partNumber`**

Lorsque vous utilisez des objets volumineux, vous pouvez avoir recours à l'en-tête HTTP `Range` pour télécharger une plage d'octets donnée à partir d'un objet. Lorsque vous utilisez l'en-tête `Range`, votre demande extrait uniquement la partie spécifiée de l'objet. Vous pouvez également utiliser l'en-tête `partNumber` pour effectuer une requête par plage pour la partie spécifiée de l'objet.

Pour plus d'informations, consultez Utilisation des en-têtes Range et partNumber.

Transformation de la date **`expiry-date`**

Vous pouvez ouvrir ou télécharger des objets transformés depuis votre point d'accès Object Lambda sur le. AWS Management Console Ces objets ne doivent pas avoir expiré. Si votre fonction Lambda transforme la date `expiry-date` de vos objets, vous pouvez voir des objets expirés qui ne peuvent pas être ouverts ou téléchargés. Ce comportement s'applique uniquement aux objets restaurés S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.

Utilisation des AWS SDK AWS CLI et

AWS Command Line Interface (AWS CLI) Les sous-commandes S3 (`cp`,`mv`, et`sync`) et l'utilisation de la AWS SDK for Java `TransferManager` classe ne sont pas prises en charge pour une utilisation avec S3 Object Lambda.

Didacticiels S3 Object Lambda

Les didacticiels suivants présentent end-to-end des procédures complètes pour certaines tâches Lambda d'objets S3.

- Didacticiel : Transformation de données pour votre application avec S3 Object Lambda
- Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend
- Tutoriel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération

Débogage de S3 Object Lambda

Les demandes adressées aux points d'accès S3 Object Lambda peuvent entraîner une nouvelle réponse d'erreur en cas de problème avec l'appel ou l'exécution de la fonction Lambda. Ces erreurs respectent le même format que les erreurs Amazon S3 standard. Pour plus d'informations sur les erreurs S3 Object Lambda, consultez la liste de codes d'erreur S3 Object Lambda dans la Référence d'API Amazon Simple Storage Service API.

Pour plus d'informations sur le débogage de fonction Lambda générale, veuillez consulter Surveillance et dépannage des applications Lambda dans le Guide du développeur AWS Lambda .

Pour plus d'informations sur les erreurs Simple Storage Service (Amazon S3) standard, consultez Réponses d'erreur dans la Référence d'API Amazon Simple Storage Service.

Vous pouvez activer les métriques de demande dans Amazon CloudWatch pour vos points d'accès Object Lambda. Ces métriques vous aident à surveiller les performances opérationnelles de votre point d'accès. Vous pouvez activer les métriques de demande pendant ou après la création de votre point d'accès Object Lambda. Pour plus d'informations, consultez Métriques de demande Lambda d'un objet S3 dans CloudWatch.

Pour obtenir une journalisation plus détaillée des demandes adressées à vos points d'accès Object Lambda, vous pouvez activer les événements de données AWS CloudTrail . Pour plus d'informations, veuillez consulter Consignation d'événements de données pour les journaux d'activité dans le Guide de l'utilisateur AWS CloudTrail .

Pour obtenir des didacticiels S3 Object Lambda, consultez :

- Didacticiel : Transformation de données pour votre application avec S3 Object Lambda
- Didacticiel : Détecter et expurger des DPI avec S3 Object Lambda et Amazon Comprehend
- Tutoriel : Utilisation de S3 Object Lambda pour filigraner dynamiquement des images au fur et à mesure de leur récupération

Pour plus d'informations sur les points d'accès standard, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Pour plus d'informations sur l'utilisation des compartiments, consultez Présentation des compartiments. Pour en savoir plus sur l'utilisation des objets, consultez Présentation des objets Amazon S3.

Qu’est-ce que S3 Express One Zone ?

Amazon S3 Express One Zone est une classe de stockage Amazon S3 à zone unique et hautes performances, spécialement conçue pour fournir un accès aux données constant en moins de dix millisecondes pour vos applications les plus sensibles à la latence. S3 Express One Zone est la classe de stockage d'objets cloud à latence la plus faible disponible à ce jour, avec des vitesses d'accès aux données jusqu'à 10 fois plus rapides et des coûts de demande 50 % inférieurs à ceux de S3 Standard. Les applications peuvent bénéficier immédiatement du fait que les demandes sont traitées jusqu'à un ordre de grandeur plus rapidement. S3 Express One Zone offre une élasticité de performance similaire à celle des autres classes de stockage S3.

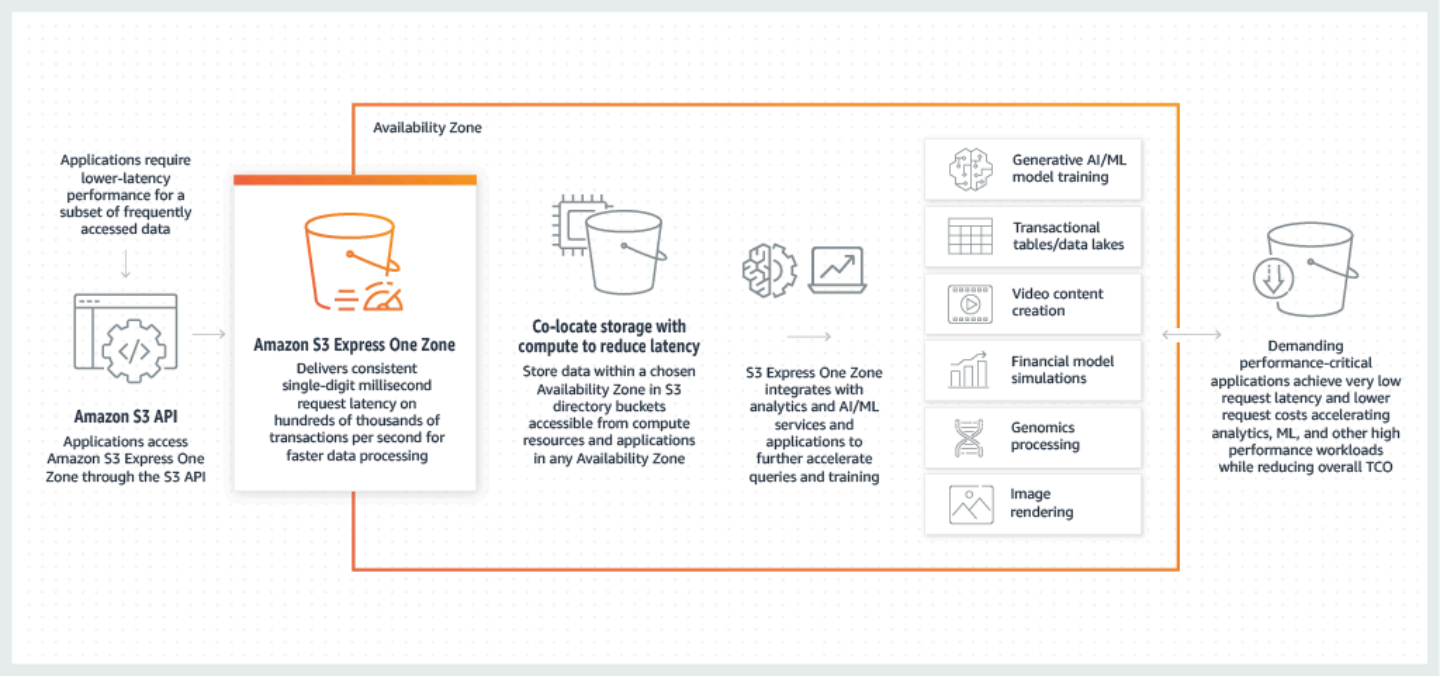
Comme pour les autres classes de stockage Amazon S3, vous n'avez pas besoin de planifier ou de provisionner à l'avance les exigences en matière de capacité ou de débit. Vous pouvez augmenter ou diminuer votre espace de stockage, en fonction des besoins, et accéder à vos données via l'API Amazon S3.

S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d’objets et les ressources de calcul, ce qui assure la vitesse d’accès la plus élevée possible. En outre, pour augmenter encore la vitesse d'accès et prendre en charge des centaines de milliers de demandes par seconde, les données de la classe de stockage S3 Express One Zone sont stockées dans un nouveau type de compartiment : un compartiment d'annuaire Amazon S3. Chaque compartiment de répertoires peut prendre en charge des centaines de milliers de transactions par seconde (TPS), quels que soient le nom de clé et le modèle d’accès.

La classe de stockage Amazon S3 Express One Zone est conçue pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenue par le contrat de niveau de service Amazon S3. Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d’une même zone de disponibilité. S3 Express One Zone est conçu pour gérer les défaillances simultanées de périphériques en détectant et réparant rapidement toute perte de redondance. En cas de panne de l’appareil existant, S3 Express One Zone transfère automatiquement les demandes vers de nouveaux appareils au sein d’une zone de disponibilité. Cette redondance permet de garantir un accès ininterrompu à vos données au sein d’une zone de disponibilité.

S3 Express One Zone est idéal pour toute application où il est important de réduire au maximum la latence requise pour accéder à un objet. Ces applications peuvent être des flux de travail interactifs, comme le montage vidéo, où les professionnels de la création ont besoin d'un accès réactif au

contenu depuis leurs interfaces utilisateur. S3 Express One Zone profite également aux charges de travail d'analytique et de machine learning qui ont des exigences similaires en matière de réactivité de leurs données, notamment aux charges de travail présentant de nombreux petits accès ou un grand nombre d'accès aléatoires. S3 Express One Zone peut être utilisé avec d'autres applications Services AWS pour prendre en charge les charges de travail d'analyse, d'intelligence artificielle et d'apprentissage automatique (AI/ML), telles qu'Amazon EMR, Amazon et SageMaker Amazon Athena.



Lorsque vous utilisez S3 Express One Zone, vous pouvez interagir avec votre compartiment de répertoire dans un cloud privé virtuel (VPC) à l'aide d'un point de terminaison VPC de passerelle. Avec un point de terminaison de passerelle, vous pouvez accéder aux compartiments de répertoire S3 Express One Zone depuis votre VPC sans passerelle Internet ni périphérique NAT pour votre VPC, et sans frais supplémentaires.

Vous pouvez utiliser la plupart des opérations et fonctionnalités de l'API Amazon S3 avec les compartiments de répertoire que vous utilisez avec les compartiments à usage général et les autres classes de stockage. Cela inclut notamment Mountpoint pour Amazon S3, le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3), les opérations par lots S3 et le blocage de l'accès public Amazon S3. Vous pouvez accéder à S3 Express One Zone à l'aide de la console Amazon S3 AWS Command Line Interface (AWS CLI), AWS des SDK et de l'API REST Amazon S3.

Pour plus d'informations sur S3 Express One Zone, consultez les rubriques suivantes.

- Présentation

Présentation

Pour optimiser les performances et réduire la latence, S3 Express One Zone introduit les nouveaux concepts suivants.

Zone de disponibilité unique

La classe de stockage Amazon S3 Express One Zone est conçue pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenue par le contrat de niveau de service Amazon S3. Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu pour gérer les défaillances simultanées de périphériques en détectant et réparant rapidement toute perte de redondance. En cas de panne de l'appareil existant, S3 Express One Zone transfère automatiquement les demandes vers de nouveaux appareils au sein d'une zone de disponibilité. Cette redondance permet de garantir un accès ininterrompu à vos données au sein d'une zone de disponibilité.

Une zone de disponibilité est un ou plusieurs centres de données discrets dotés d'une alimentation, d'un réseau et d'une connectivité redondants dans une Région AWS. Lorsque vous créez un bucket d'annuaire, vous choisissez la zone de disponibilité et Région AWS l'emplacement de votre bucket.

Compartiments de répertoires

Il existe deux types de compartiments Amazon S3 : les compartiments S3 à usage général et les compartiments d'annuaire S3. Les compartiments à usage général constituent le type de compartiment Amazon S3 par défaut utilisé dans la grande majorité des cas d'utilisation de S3. Les compartiments de répertoires utilisent uniquement la classe de stockage S3 Express One Zone, conçue pour les charges de travail ou les applications critiques en termes de performances qui nécessitent une latence constante inférieure à dix millisecondes. Choisissez le type de godet qui correspond le mieux à votre application et à vos exigences de performance.

Les compartiments d'annuaire organisent les données de manière hiérarchique dans des répertoires, contrairement à la structure de stockage plate des compartiments à usage général. Il n'existe pas

de limite de préfixe pour les compartiments de répertoires et les répertoires individuels peuvent faire l'objet d'une mise à l'échelle horizontale.

Les compartiments de répertoires utilisent la classe de stockage S3 Express One Zone, conçue pour être utilisée par des applications sensibles aux performances. Avec S3 Express One Zone, vous pouvez sélectionner une seule zone de disponibilité avec l'option de colocaliser votre stockage d'objets avec vos ressources informatiques, ce qui fournit la vitesse d'accès la plus élevée possible. Cela ne ressemble pas aux compartiments à usage général, dans lesquels des objets sont stockés de manière redondante dans plusieurs zones de disponibilité. Régions AWS

Pour plus d'informations sur les compartiments de répertoires, consultez Compartiments de répertoire. Pour plus d'informations sur les compartiments à usage général, consultez Présentation des compartiments.

Points de terminaison et points de terminaison de VPC de passerelle

Les opérations d'API de gestion de compartiments pour les compartiments d'annuaire sont disponibles via un point de terminaison régional et sont appelées opérations d'API de point de terminaison régional. `CreateBucket` et `DeleteBucket` sont des exemples d'opérations d'API de point de terminaison régional. Après avoir créé un compartiment de répertoires, vous pouvez utiliser les opérations d'API de point de terminaison zonal pour charger et gérer les objets dans votre compartiment de répertoires. Les opérations d'API de point de terminaison zonal sont disponibles via un point de terminaison zonal. `PutObject` et `CopyObject` sont des exemples d'opérations d'API de point de terminaison zonal.

Vous pouvez accéder à S3 Express One Zone depuis votre VPC en utilisant les points de terminaison VPC de passerelle. Après avoir créé un point de terminaison de passerelle, vous pouvez l'ajouter comme cible dans votre table de routage pour le trafic destiné à S3 Express One Zone depuis votre VPC. Comme avec Amazon S3, aucuns frais supplémentaires ne s'appliquent à l'utilisation de points de terminaison de passerelle. Pour plus d'informations sur la manière de configurer des points de terminaison de VPC de passerelle, consultez Mise en réseau pour S3 Express One Zone.

Autorisation basée sur les sessions

Avec S3 Express One Zone, vous authentifiez et autorisez les demandes par le biais d'un nouveau mécanisme basé sur les sessions, optimisé pour fournir une latence minimale. Vous pouvez utiliser `CreateSession` pour demander des informations d'identification temporaires afin de bénéficier d'un accès à faible latence à votre compartiment. Ces informations d'identification temporaires sont

limitées à un compartiment de répertoires S3 spécifique. Les jetons de session ne sont utilisés qu'avec les opérations zonales (au niveau de l'objet) (à l'exception de). CopyObject Pour plus d'informations, consultez Autorisation `CreateSession`.

Les AWS SDK pris en charge pour S3 Express One Zone gèrent l'établissement et le rafraîchissement des sessions en votre nom. Pour protéger vos sessions, les informations d'identification de sécurité temporaires expirent au bout de 5 minutes. Après avoir téléchargé et installé les AWS SDK et configuré les autorisations AWS Identity and Access Management (IAM) nécessaires, vous pouvez immédiatement commencer à utiliser les opérations d'API.

Fonctionnalités de S3 Express One Zone

Les fonctionnalités S3 suivantes sont disponibles pour S3 Express One Zone. Pour obtenir la liste complète des opérations d'API prises en charge et des fonctionnalités non prises en charge, consultez. En quoi S3 Express One Zone est-il différent ?

Gestion des accès et sécurité

Avec les compartiments de répertoires, vous pouvez utiliser les fonctionnalités suivantes pour auditer et gérer l'accès. Par défaut, les compartiments de répertoires sont privés et sont accessibles uniquement par les utilisateurs auxquels l'accès a été explicitement accordé. Contrairement aux compartiments à usage général, qui peuvent définir la limite de contrôle d'accès au niveau des compartiments, des préfixes ou des balises d'objet, la limite de contrôle d'accès pour les compartiments de répertoires est définie uniquement au niveau des compartiments. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

- Accès public au bloc S3 : tous les paramètres d'accès public au bloc S3 sont activés par défaut au niveau du compartiment. Ce paramètre par défaut ne peut pas être modifié.
- Propriété des objets S3 (propriétaire du compartiment appliqué par défaut) : les listes de contrôle d'accès (ACL) ne sont pas prises en charge pour les compartiments d'annuaire. Les compartiments de répertoire utilisent automatiquement le paramètre imposé par le propriétaire du compartiment pour la propriété des objets S3. L'application par le propriétaire du compartiment signifie que les ACL sont désactivées et que le propriétaire du compartiment possède automatiquement tous les objets du compartiment et en a le contrôle total. Ce paramètre par défaut ne peut pas être modifié.
- AWS Identity and Access Management (IAM) — IAM vous aide à contrôler en toute sécurité l'accès à vos compartiments d'annuaire. Vous pouvez utiliser IAM pour accorder l'accès aux opérations d'API de gestion des compartiments (régionales) et aux opérations d'API de gestion

des objets (Zonal) par le biais de l'`s3express:CreateSession`action. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone. Contrairement aux actions de gestion d'objets, les actions de gestion de compartiments ne peuvent pas être effectuées entre comptes. Seul le propriétaire du compartiment peut effectuer ces actions.

- Politiques de compartiment : utilisez un langage de politique basé sur IAM afin de configurer les autorisations basées sur les ressources pour vos compartiments de répertoires. Vous pouvez également utiliser IAM pour contrôler l'accès aux opérations d'`CreateSession`API, ce qui vous permet d'utiliser les opérations d'API zonales, ou de gestion d'objets. Vous pouvez accorder l'accès à un même compte ou à plusieurs comptes aux opérations de l'API Zonal. Pour plus d'informations sur les autorisations et les politiques de S3 Express One Zone, consultezAWS Identity and Access Management (IAM) pour S3 Express One Zone.
- Analyseur d'accès IAM pour S3 : évaluez et surveillez vos politiques d'accès pour vous assurer qu'elles fournissent uniquement l'accès prévu à vos ressources S3.

Journalisation et surveillance

S3 Express One Zone utilise les outils de journalisation et de surveillance S3 suivants que vous pouvez utiliser pour surveiller et contrôler l'utilisation de vos ressources :

- Amazon CloudWatch Metrics — Surveillez vos AWS ressources et vos applications en les utilisant CloudWatch pour collecter et suivre les métriques. S3 Express One Zone utilise le même espace de CloudWatch noms que les autres classes de stockage Amazon S3 (`AWS/S3`) et prend en charge les métriques de stockage quotidiennes pour les compartiments de répertoire : `BucketSizeBytes` et. `NumberOfObjects` Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.
- AWS CloudTrail logs — AWS CloudTrail est un outil Service AWS qui vous aide à mettre en œuvre l'audit opérationnel et des risques, la gouvernance et la conformité de votre entreprise Compte AWS en enregistrant les actions entreprises par un utilisateur, un rôle ou un Service AWS. Pour S3 Express One Zone, CloudTrail capture les opérations de l'API du point de terminaison régional (par exemple, `CreateBucket` et`PutBucketPolicy`) en tant qu'événements de gestion. Ces événements incluent les actions entreprises dans le cadre des opérations AWS Management Console, AWS Command Line Interface (AWS CLI), AWS des SDK et des AWS API. Les événements CloudTrail de gestion `eventsource` pour S3 Express One Zone sont`s3express.amazonaws.com`. Pour plus d'informations, consultez CloudTrail Événements Amazon S3.

> **Note**
>
> Les journaux d'accès au serveur Amazon S3 ne sont pas pris en charge avec S3 Express One Zone.

Gestion des objets

Après avoir créé un compartiment de répertoire, vous pouvez gérer votre stockage d'objets à l'aide de la console Amazon S3, AWS des SDK et AWS CLI. Les fonctionnalités suivantes sont disponibles pour la gestion des objets avec S3 Express One Zone :

- Opérations par lots S3 : utilisez les opérations par lots pour effectuer des opérations groupées sur des objets dans des compartiments de répertoire, par exemple, la AWS Lambda fonction Copy and Invoke. Par exemple, vous pouvez utiliser les opérations par lots pour copier des objets entre des compartiments de répertoires et des compartiments à usage général. Avec Batch Operations, vous pouvez gérer des milliards d'objets à grande échelle avec une seule requête S3 à l'aide AWS des SDK AWS CLI ou en quelques clics dans la console Amazon S3.
- Importation : après avoir créé un compartiment de répertoires, vous pouvez le remplir d'objets à l'aide de la fonctionnalité d'importation dans la console Amazon S3. L'importation est une méthode simplifiée pour créer des tâches d'opérations par lots afin de copier des objets depuis des compartiments à usage général vers des compartiments de répertoires.

AWS SDK et bibliothèques clientes

Après avoir créé un compartiment de répertoire et chargé un objet dans celui-ci, vous pouvez gérer votre stockage d'objets à l'aide des méthodes suivantes.

- Mountpoint pour Amazon S3 — Mountpoint pour Amazon S3 est un client de fichiers open source qui fournit un accès haut débit, réduisant ainsi les coûts de calcul pour les lacs de données sur Amazon S3. Mountpoint pour Amazon S3 traduit les appels d'API du système de fichiers local en appels d'API d'objets S3 tels que `GET` et. `LIST` Il est idéal pour les charges de travail de lacs de données à lecture intense qui traitent des pétaoctets de données et ont besoin du débit élastique élevé fourni par Amazon S3 pour augmenter ou diminuer le volume sur des milliers d'instances.
- S3A— S3A est une interface Hadoop compatible recommandée pour accéder aux magasins de données dans Amazon S3. S3Aremplace le client S3N Hadoop du système de fichiers.

- PyTorchon AWS — PyTorch on AWS est un framework d'apprentissage profond open source qui facilite le développement de modèles d'apprentissage automatique et leur déploiement en production.
- AWS SDK — Vous pouvez utiliser les AWS SDK lorsque vous développez des applications avec Amazon S3. Les AWS SDK simplifient vos tâches de programmation en encapsulant l'API REST Amazon S3 sous-jacente. Pour plus d'informations sur l'utilisation AWS des kits SDK avec S3 Express One Zone, consultezthe section called “AWS SDK”.

Chiffrement et protection des données

Les objets stockés dans des compartiments de répertoire sont automatiquement chiffrés à l'aide du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3). Les compartiments d'annuaire ne prennent pas en charge le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C) ou le chiffrement double couche côté serveur avec (DSSE-KMS). AWS KMS keys Pour plus d’informations, consultez Protection et chiffrement des données et Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

S3 Express One Zone vous offre la possibilité de choisir l’algorithme de somme de contrôle utilisé pour valider vos données pendant le chargement ou le téléchargement. Vous pouvez sélectionner l’un des algorithmes de contrôle d’intégrité des données Secure Hash Algorithms (SHA) ou Cyclic Redundancy Check (CRC) suivants : CRC32, CRC32C, SHA-1 ou SHA-256. Les checksums basés sur MD5 ne sont pas pris en charge avec la classe de stockage S3 Express One Zone.

Pour plus d’informations, consultez Bonnes pratiques supplémentaires en matière de somme de contrôle S3.

AWS Version de signature (4SigV4)

S3 Express One Zone utilise AWS la version de signature 4 (SigV4). SigV4est un protocole de signature utilisé pour authentifier les demandes adressées à Amazon S3 via HTTPS. S3 Express One Zone signe les demandes en utilisant AWS Sigv4. Pour plus d'informations, consultez Authentification des demandes (AWS Signature version 4) dans le manuel Amazon Simple Storage Service API Reference.

Forte cohérence

S3 Express One Zone assure une forte read-after-write cohérence pour l'ensemble des objets contenus dans vos compartiments de répertoire PUT et pour les DELETE Régions AWS requêtes y afférentes. Pour plus d'informations, consultez Modèle de cohérence des données Amazon S3.

Services connexes

Vous pouvez utiliser ce qui suit Services AWS avec la classe de stockage S3 Express One Zone pour prendre en charge votre cas d'utilisation spécifique à faible latence.

- Amazon Elastic Compute Cloud (Amazon EC2) — Amazon EC2 fournit une capacité de calcul sécurisée et évolutive dans le. AWS Cloud En utilisant Amazon EC2, vous n'avez pas besoin d'investir dans du matériel au départ, ce qui vous permet de développer et de déployer des applications plus rapidement. Vous pouvez utiliser Amazon EC2 pour lancer autant de serveurs virtuels que nécessaire, configurer la sécurité et les réseaux, et gérer le stockage.
- AWS Lambda : Lambda est un service de calcul qui vous permet d'exécuter du code sans provisionner ni gérer de serveurs. Vous configurez des paramètres de notification sur un compartiment et accordez à Amazon S3 l'autorisation d'appeler une fonction sur la stratégie d'autorisations basée sur une ressource de la fonction.
- Amazon Elastic Kubernetes Service (Amazon EKS) — Amazon EKS est un service géré qui élimine le besoin d'installer, d'exploiter et de maintenir Kubernetes votre propre plan de contrôle. AWSKubernetesest un système open source qui automatise la gestion, le dimensionnement et le déploiement des applications conteneurisées.
- Amazon Elastic Container Service (Amazon ECS) : Amazon ECS est un service d'orchestration de conteneurs entièrement géré qui vous permet de déployer, de gérer et de mettre à l'échelle aisément des applications conteneurisées.
- Amazon Athena : Amazon Athena est un service de requête interactif qui facilite l'analyse des données directement dans Amazon S3 à l'aide du langage SQL standard. Vous pouvez également utiliser Athena pour exécuter des analyses de données de manière interactive Apache Spark sans avoir à planifier, configurer ou gérer les ressources. Lorsque vous exécutez Apache Spark des applications sur Athena, vous soumettez le Spark code à traiter et vous recevez directement les résultats.
- Amazon SageMaker Runtime Model Training — Amazon SageMaker Runtime est un service d'apprentissage automatique entièrement géré. Avec SageMaker Runtime, les data scientists et les développeurs peuvent rapidement et facilement créer et entraîner des modèles d'apprentissage

automatique, puis les déployer directement dans un environnement hébergé prêt pour la production.

- AWS Glue— AWS Glue est un service d'intégration de données sans serveur qui permet aux utilisateurs d'outils d'analyse de découvrir, de préparer, de déplacer et d'intégrer facilement des données provenant de sources multiples. Vous pouvez l'utiliser AWS Glue pour l'analyse, l'apprentissage automatique et le développement d'applications. AWS Glue inclut également des outils de productivité et d'exploitation des données supplémentaires pour la création, l'exécution de tâches et la mise en œuvre de flux de travail commerciaux.
- Amazon EMR — Amazon EMR est une plate-forme de clusters gérés qui simplifie l'exécution de frameworks de mégadonnées, tels que le traitement Apache Hadoop et Apache Spark l'analyse AWS de grandes quantités de données.

Étapes suivantes

Pour plus d'informations sur l'utilisation des compartiments de répertoires et de la classe de stockage S3 Express One Zone, consultez les rubriques suivantes :

- En quoi S3 Express One Zone est-il différent ?
- Bien démarrer avec S3 Express One Zone
- Mise en réseau pour S3 Express One Zone
- Compartiments de répertoire
- Utilisation d'objets dans un bucket de répertoire
- Sécurité pour S3 Express One Zone
- Optimisation des performances d'Amazon S3 Express One Zone
- Développement avec S3 Express One Zone

En quoi S3 Express One Zone est-il différent ?

Amazon S3 Express One Zone est une classe de stockage Amazon S3 à zone unique et hautes performances, spécialement conçue pour fournir un accès aux données constant en moins de dix millisecondes pour vos applications les plus sensibles à la latence. S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d'objets et les ressources de calcul, ce qui assure la vitesse d'accès la plus élevée possible. En outre, pour augmenter encore la vitesse d'accès et

prendre en charge des centaines de milliers de demandes par seconde, les données S3 Express One Zone sont stockées dans un nouveau type de compartiment : un compartiment de répertoires Amazon S3.

Pour plus d'informations, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Vous pouvez créer des compartiments de répertoires et accéder à vos données dans S3 Express One Zone à l'aide de l'API Amazon S3. L'API Amazon S3 est compatible avec S3 Express One Zone et les compartiments de répertoires, à l'exception de quelques différences notables. Pour plus d'informations sur les différences de S3 Express One Zone, consultez les rubriques suivantes.

Rubriques

- Différences de S3 Express One Zone
- Opérations d'API prises en charge par S3 Express One Zone
- Fonctionnalités Amazon S3 non prises en charge par S3 Express One Zone

Différences de S3 Express One Zone

- Type de compartiment pris en charge : les objets de la classe de stockage S3 Express One Zone peuvent être stockés uniquement dans des compartiments de répertoires. Pour plus d'informations, consultez Compartiments de répertoire.
- Durabilité : avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu pour garantir une disponibilité de 99,95 % au sein d'une même zone de disponibilité et s'appuie sur le contrat de niveau de service Amazon S3. Pour plus d'informations, consultez Zone de disponibilité unique.
- **`ListObjectsV2`**comportement
  - Pour les compartiments de répertoire, `ListObjectsV2` ne renvoie pas les objets par ordre lexicographique (alphabétique). De plus, les préfixes doivent se terminer par un délimiteur et seul « / » peut être spécifié comme délimiteur.
  - Pour les compartiments de répertoire, la `ListObjectsV2` réponse inclut les préfixes liés uniquement aux téléchargements partitionnés en cours.
- Comportement de suppression : lorsque vous supprimez un objet dans un compartiment de répertoires, Amazon S3 supprime de manière récursive tous les répertoires vides au niveau du

chemin de l'objet. Par exemple, si vous supprimez la clé d'objet`dir1/dir2/file1.txt`, Amazon S3 la supprime`file1.txt`. Si les répertoires `dir1/` et `dir2/` sont vides et ne contiennent aucun autre objet, Amazon S3 supprime également ces répertoires.

- ETags et sommes de contrôle : les balises d'entité (ETags) pour S3 Express One Zone sont des chaînes alphanumériques aléatoires et non pas des sommes de contrôle MD5. Pour plus d'informations sur l'utilisation de sommes de contrôle supplémentaires avec S3 Express One Zone, consultez Bonnes pratiques supplémentaires en matière de somme de contrôle S3.
- Clés d'objet dans les demandes **`DeleteObjects`**
  - Les clés d'objet figurant dans les demandes `DeleteObjects` doivent contenir au moins un caractère autre qu'une espace. Les chaînes composées uniquement de caractères espace ne sont pas prises en charge dans les demandes `DeleteObjects`.
  - Les clés d'objet figurant dans les demandes `DeleteObjects` ne peuvent pas contenir de caractères de contrôle Unicode, à l'exception des caractères de saut de ligne (`\n`), de tabulation (`\t`) et de retour chariot (`\r`).
- Points de terminaison régionaux et zonaux : lorsque vous utilisez S3 Express One Zone, vous devez spécifier la région dans toutes les demandes client. Pour les points de terminaison régionaux, vous spécifiez la région, par exemple `s3express-control.us-west-2.amazonaws.com`. Pour les points de terminaison zonaux, vous spécifiez à la fois la région et la zone de disponibilité, par exemple `s3express-usw2-az1.us-west-2.amazonaws.com`. Pour plus d'informations, consultez Points de terminaison régionaux et zonaux.
- Chargements partitionnés : comme avec les autres objets stockés dans Amazon S3, vous pouvez charger et copier des objets volumineux stockés dans la classe de stockage S3 Express One Zone en utilisant le processus de chargement partitionné. Toutefois, voici quelques différences lors de l'utilisation du processus de chargement partitionné avec des objets stockés dans S3 Express One Zone. Pour plus d'informations, consultez the section called "Utilisation de téléchargements partitionnés avec des compartiments de répertoires".
  - La date de création de l'objet correspond à la date d'achèvement du chargement partitionné.
  - Les numéros de parties partitionnés doivent utiliser des numéros de parties consécutifs. Si vous essayez d'effectuer une demande de chargement partitionné avec des numéros de parties non consécutifs, Amazon S3 génère une erreur HTTP `400 (Bad Request)`.
  - L'initiateur d'un chargement partitionné ne peut abandonner la demande de chargement partitionné que s'il a obtenu une autorisation d'accès explicite à `AbortMultipartUpload` via l'autorisation `s3express:CreateSession`. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

- Videz un compartiment de répertoire : la `s3 rm` commande via la AWS Command Line Interface (CLI), l'`delete`opération via Mountpoint et le bouton d'option Empty bucket via le ne permettent pas de supprimer les téléchargements AWS Management Console partitionnés en cours dans un compartiment de répertoire. Pour supprimer ces téléchargements partitionnés en cours, utilisez l'`ListMultipartUploads`opération pour répertorier les téléchargements partitionnés en cours dans le bucket et utilisez l'`AbortMultupartUpload`opération pour abandonner tous les téléchargements partitionnés en cours.

Opérations d'API prises en charge par S3 Express One Zone

La classe de stockage Amazon S3 Express One Zone prend en charge les opérations d'API de point de terminaison régional (niveau compartiment ou plan de contrôle) et zonal (niveau objet ou plan de données). Pour plus d'informations, consultez Mise en réseau pour S3 Express One Zone et Points de terminaison et points de terminaison de VPC de passerelle.

Opérations d'API de point de terminaison régional

Les opérations d'API de point de terminaison régional suivantes sont prises en charge pour S3 Express One Zone :

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Opérations d'API de point de terminaison zonal

Les opérations d'API de point de terminaison zonal suivantes sont prises en charge pour S3 Express One Zone :

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects

- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy

Fonctionnalités Amazon S3 non prises en charge par S3 Express One Zone

Les fonctionnalités Amazon S3 suivantes ne sont pas prises en charge par S3 Express One Zone :

- AWS CloudTrail événements du plan de données
- AWS politiques gérées
- AWS PrivateLink pour S3
- Sommes de contrôle MD5
- Suppression de l'authentification multifacteur (MFA)
- Verrouillage des objets S3
- paiement par le demandeur
- Octrois d’accès S3
- Points d’accès S3
- Balises de compartiment
- Métriques relatives aux CloudWatch demandes Amazon
- Notifications d'événements S3

- Cycle de vie S3
- Points d’accès multirégionaux S3
- Points d’accès S3 Object Lambda
- Gestion des versions S3
- Inventaire S3
- Réplication S3
- Balises d’objet
- S3 Select
- Journaux d’accès de serveur
- Hébergement de site Web statique
- S3 Storage Lens
- Groupes S3 Storage Lens
- S3 Transfer Acceleration
- Chiffrement double couche côté serveur avec clés AWS Key Management Service (AWS KMS) (DSSE-KMS)
- Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS)
- Chiffrement côté serveur avec clés fournies par le client (SSE-C)
- L'option permettant de copier les paramètres d'un compartiment existant lors de la création d'un nouveau compartiment dans AWS Management Console.

Bien démarrer avec S3 Express One Zone

La section suivante explique comment commencer à utiliser la classe de stockage et les compartiments de répertoires Amazon S3 Express One Zone. Pour plus d’informations, consultez Qu’est-ce que S3 Express One Zone ?.

Rubriques

- Configuration AWS Identity and Access Management (IAM) avec S3 Express One Zone
- Configuration de points de terminaison de VPC de passerelle
- Travaillez avec S3 Express One Zone à l'aide de la console S3 et des AWS SDK AWS CLI

Configuration AWS Identity and Access Management (IAM) avec S3 Express One Zone

AWS Identity and Access Management (IAM) est un outil Service AWS qui aide les administrateurs à contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (disposer d'autorisations) à utiliser des ressources Amazon S3 dans S3 Express One Zone. Vous pouvez utiliser IAM sans frais supplémentaires.

Par défaut, les utilisateurs ne disposent pas d'autorisations pour les compartiments de répertoires et les opérations S3 Express One Zone. Pour accorder des autorisations d'accès pour les compartiments de répertoires et les opérations S3 Express One Zone, vous pouvez utiliser IAM pour créer des utilisateurs ou des rôles et associer des autorisations à ces identités.

Pour bien démarrer avec IAM, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone et Politiques basées sur l'identité IAM pour S3 Express One Zone.

Configuration de points de terminaison de VPC de passerelle

Pour accéder à S3 Express One Zone, vous utilisez des points de terminaison régionaux et zonaux différents des points de terminaison Amazon S3 standard. Selon l'opération d'API Amazon S3 que vous utilisez, un point de terminaison zonal ou régional est requis. Pour obtenir la liste complète des opérations d'API prises en charge par type de point de terminaison, consultez Opérations d'API prises en charge par S3 Express One Zone. Vous devez accéder aux points de terminaison zonaux et régionaux via un point de terminaison de cloud privé virtuel (VPC) de passerelle. Pour configurer des points de terminaison de passerelle, consultez Mise en réseau pour S3 Express One Zone.

Travaillez avec S3 Express One Zone à l'aide de la console S3 et des AWS SDK AWS CLI

Vous pouvez utiliser la classe de stockage et les compartiments de répertoire S3 Express One Zone à l'aide des AWS SDK, de la console Amazon S3 AWS Command Line Interface (AWS CLI) et de l'API REST Amazon S3.

Console S3

Pour bien démarrer avec la console S3, effectuez les étapes suivantes :

- Création d'un compartiment de répertoires

- Vidage d’un compartiment de répertoires
- Suppression d’un compartiment de répertoires

AWS SDK

S3 Express One Zone prend en charge les AWS SDK suivants :

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java 2.x
- AWS SDK for JavaScript v3
- AWS SDK for .NET
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby
- Kit AWS SDK pour Kotlin
- Kit AWS SDK pour Rust

Lorsque vous utilisez S3 Express One Zone, nous vous recommandons d’utiliser la dernière version des kits AWS SDK. Les AWS SDK pris en charge pour S3 Express One Zone gèrent l'établissement, le rafraîchissement et la résiliation des sessions en votre nom. Cela signifie que vous pouvez immédiatement commencer à utiliser les opérations d'API après avoir téléchargé et installé les AWS SDK et configuré les autorisations IAM nécessaires. Pour de plus amples informations, veuillez consulter AWS Identity and Access Management (IAM) pour S3 Express One Zone.

Pour plus d'informations sur les AWS SDK, notamment sur la façon de les télécharger et de les installer, consultez la section Outils sur AWS lesquels vous pouvez vous appuyer.

Pour des exemples de AWS SDK, consultez ce qui suit :

- Création d’un compartiment de répertoires
- Vidage d’un compartiment de répertoires
- Suppression d’un compartiment de répertoires

AWS Command Line Interface (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer des compartiments de répertoires et utiliser les opérations d'API de point de terminaison régionales et zonales prises en charge pour S3 Express One Zone.

Pour commencer à utiliser le AWS CLI, voir Commencer avec le AWS CLI dans le manuel de référence des AWS CLI commandes.

> **Note**
>
> Pour utiliser des compartiments de répertoire avec les `aws s3` commandes de haut niveau, installez AWS CLI la dernière version. Pour plus d'informations sur l'installation et la configuration du AWS CLI, voir Installer ou mettre à jour la dernière version du AWS CLI dans le manuel de référence des AWS CLI commandes.

Pour AWS CLI des exemples, consultez ce qui suit :

- Création d'un compartiment de répertoires
- Vidage d'un compartiment de répertoires
- Suppression d'un compartiment de répertoires

Mise en réseau pour S3 Express One Zone

Pour accéder aux compartiments de répertoires et aux objets de classe de stockage Amazon S3 Express One Zone, vous utilisez des points de terminaison d'API régionaux et zonaux différents des points de terminaison Amazon S3 standard. Selon l'opération d'API S3 que vous utilisez, un point de terminaison zonal ou régional est requis. Pour obtenir la liste complète des opérations d'API par type de point de terminaison, consultez Opérations d'API prises en charge par S3 Express One Zone.

Vous pouvez accéder aux opérations d'API zonales et régionales via des points de terminaison de cloud privé virtuel (VPC) de passerelle. Pour configurer des points de terminaison de VPC de passerelle, consultez the section called "Configuration de points de terminaison de VPC de passerelle".

Les rubriques suivantes décrivent les exigences de mise en réseau pour accéder à S3 Express One Zone à l'aide d'un point de terminaison de VPC de passerelle.

Rubriques

Points de terminaison

Vous pouvez accéder aux compartiments de répertoires et aux objets de classe de stockage Amazon S3 Express One Zone à partir de votre VPC en utilisant des points de terminaison de VPC de passerelle. S3 Express One Zone utilise des points de terminaison d'API régionaux et zonaux. Selon l'opération d'API Amazon S3 que vous utilisez, un point de terminaison régional ou zonal est requis. Il n'y a pas de frais supplémentaires pour l'utilisation de points de terminaison de passerelle.

Les opérations d'API de niveau compartiment (ou plan de contrôle) sont disponibles via les points de terminaison régionaux et sont appelées opérations d'API de point de terminaison régional. `CreateBucket` et `DeleteBucket` sont des exemples d'opérations d'API de point de terminaison régional. Lorsque vous créez un compartiment de répertoires, vous choisissez une zone de disponibilité unique dans laquelle votre compartiment de répertoires sera créé. Après avoir créé un compartiment de répertoires, vous pouvez utiliser les opérations d'API de point de terminaison zonal pour charger et gérer les objets dans votre compartiment de répertoires.

Les opérations d'API de niveau objet (ou plan de données) sont disponibles via les points de terminaison zonaux et sont appelées opérations d'API de point de terminaison zonal. `CreateSession` et `PutObject` sont des exemples d'opérations d'API de point de terminaison zonal.

Le tableau suivant indique les points de terminaison d'API régionaux et zonaux disponibles pour chaque région et zone de disponibilité.

Configuration de points de terminaison de VPC de passerelle

Utilisez la procédure suivante pour créer un point de terminaison de passerelle qui se connecte à des compartiments de répertoires et des objets de classe de stockage Amazon S3 Express One Zone.

Pour configurer un point de terminaison de VPC de passerelle

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le panneau de navigation, choisissez Points de terminaison.
3. Choisissez Créer un point de terminaison.

4. Créez un nom pour votre point de terminaison.
5. Pour Service category (Catégorie de service), choisissez Services AWS.
6. Pour Services, ajoutez le filtre Type=Passerelle, puis choisissez le bouton d'option à côté de com.amazonaws.***région***.s3express.
7. Pour VPC, choisissez le VPC dans lequel créer le point de terminaison.
8. Pour Configure route tables (Configurer les tables de routage), sélectionnez les tables de routage qui seront utilisées par le point de terminaison. Amazon VPC ajoute automatiquement une route pour pointer le trafic destiné au service vers l'interface réseau du point de terminaison.
9. Pour Politique, choisissez Accès complet afin d'autoriser toutes les opérations de tous les principaux sur toutes les ressources via le point de terminaison de VPC. Sinon, choisissez Personnalisé pour attacher une politique de point de terminaison de VPC qui contrôle les autorisations dont disposent les principaux pour effectuer des actions sur les ressources via le point de terminaison de VPC.
10. (Facultatif) Pour ajouter une balise, choisissez Ajouter une nouvelle balise et entrez la clé et la valeur de la balise.
11. Choisissez Créer un point de terminaison.

Après avoir créé un point de terminaison de passerelle, vous pouvez utiliser les points de terminaison d'API régionaux et les points de terminaison d'API zonaux pour accéder aux compartiments de répertoires et aux objets de classe de stockage Amazon S3 Express One Zone.

Compartiments de répertoire

Il existe deux types de compartiments Amazon S3 : les compartiments à usage général et les compartiments de répertoires. Choisissez le type de compartiment qui correspond le mieux à votre application et à vos exigences de performances :

- Les compartiments à usage général constituent le type de compartiment S3 d'origine et sont recommandés pour la plupart des cas d'utilisation et des modèles d'accès. Les compartiments à usage général autorisent également les objets stockés dans toutes les classes de stockage, à l'exception de S3 Express One Zone.
- Les compartiments de répertoires utilisent la classe de stockage S3 Express One Zone, recommandée si votre application est sensible aux performances et peut bénéficier d'une latence inférieure à dix millisecondes pour les demandes PUT et GET.

Les compartiments de répertoires sont utilisés pour les charges de travail ou les applications critiques en termes de performances qui nécessitent une latence constante inférieure à dix millisecondes. Les compartiments de répertoires organisent les données de manière hiérarchique dans des répertoires, contrairement à la structure de stockage horizontale des compartiments à usage général. Il n'existe pas de limite de préfixe pour les compartiments de répertoires et les répertoires individuels peuvent faire l'objet d'une mise à l'échelle horizontale.

Les compartiments de répertoires utilisent la classe de stockage S3 Express One Zone, qui stocke les données sur plusieurs appareils au sein d'une même zone de disponibilité, mais ne stocke pas les données de manière redondante entre les zones de disponibilité. Lorsque vous créez un bucket d'annuaire, nous vous recommandons de spécifier une zone de disponibilité Région AWS et une zone de disponibilité locales pour vos instances de calcul Amazon EC2, Amazon Elastic Kubernetes Service ou Amazon Elastic Container Service (Amazon ECS) afin d'optimiser les performances.

Vous pouvez créer jusqu'à 10 compartiments de répertoire dans chacun de vos répertoires Comptes AWS, sans limite quant au nombre d'objets que vous pouvez stocker dans un compartiment. Votre quota de compartiments est appliqué à chaque région dans votre Compte AWS. Si votre demande nécessite une augmentation de cette limite, contactez AWS Support. Pour plus d'informations, consultez la console Service Quotas.

> **Important**
>
> Les compartiments d'annuaire qui n'ont aucune activité de demande pendant une période d'au moins 90 jours passent à l'état inactif. Lorsqu'il est dans un état inactif, un compartiment de répertoires est temporairement inaccessible pour les lectures et les écritures. Les compartiments inactifs conservent tout le stockage, ainsi que toutes les métadonnées d'objet et de compartiment. Les frais de stockage existants s'appliquent aux compartiments inactifs. Si vous faites une demande d'accès à un compartiment inactif, celui-ci passe à l'état actif, généralement en quelques minutes. Pendant cette période de transition, les lectures et les écritures renvoient un code `503 (Service Unavailable)` d'erreur HTTP.

Les rubriques suivantes fournissent des informations sur les compartiments de répertoires. Pour plus d'informations sur les compartiments à usage général, consultez Présentation des compartiments.

Rubriques

- Zones de disponibilité
- Noms des compartiments de répertoires

- Annuaires
- Noms de clés
- Gestion des accès
- Utilisation des compartiments de répertoires
- Règles de dénomination des compartiments de répertoires
- Création d'un compartiment de répertoires
- Affichage des propriétés d'un compartiment de répertoires
- Gestion des politiques de compartiment pour les compartiments de répertoires
- Vidage d'un compartiment de répertoires
- Suppression d'un compartiment de répertoires
- Établissement de la liste des compartiments de répertoires
- Utilisation HeadBucket avec des buckets de répertoires

Zones de disponibilité

Lorsque vous créez un compartiment de répertoires, vous choisissez la zone de disponibilité et la Région AWS.

Les compartiments de répertoires utilisent la classe de stockage S3 Express One Zone, conçue pour être utilisée par des applications sensibles aux performances. S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d'objets et les ressources de calcul, ce qui assure la vitesse d'accès la plus élevée possible.

Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenu par le contrat de niveau de service Amazon S3. Pour plus d'informations, consultez Zone de disponibilité unique.

Noms des compartiments de répertoires

Le nom d'un compartiment de répertoires se compose d'un nom de base que vous fournissez et d'un suffixe contenant l'ID de la zone de disponibilité où se trouve votre compartiment. Les noms de compartiments de répertoires doivent utiliser le format suivant et respecter les règles de dénomination des compartiments de répertoires :

```
bucket-base-name--azid--x-s3
```

Par exemple, le nom de compartiment de répertoires suivant contient l'ID de zone de disponibilité `usw2-az1` :

```
bucket-base-name--usw2-az1--x-s3
```

Pour plus d'informations, consultez Règles de dénomination des compartiments de répertoires.

Annuaires

Les compartiments de répertoires organisent les données de manière hiérarchique dans des répertoires, contrairement à la structure de tri horizontale des compartiments à usage général. Chaque compartiment de répertoires S3 peut prendre en charge des centaines de milliers de transactions par seconde (TPS), indépendamment du nombre de répertoires qu'il contient.

Avec un espace de noms hiérarchique, le délimiteur figurant dans la clé d'objet est important. Le seul délimiteur pris en charge est la barre oblique (`/`). Les répertoires sont déterminés par les limites des délimiteurs. Par exemple, la clé d'objet `dir1/dir2/file1.txt` entraîne la création automatique des répertoires `dir1/` et `dir2/`, et l'ajout de l'objet `file1.txt` dans le répertoire `/dir2`, dans le chemin `dir1/dir2/file1.txt`.

Le modèle d'indexation des compartiments de répertoires renvoie des résultats non triés pour l'opération d'API `ListObjectsV2`. Si vous devez limiter vos résultats à une sous-section de votre compartiment, vous pouvez spécifier un chemin de sous-répertoire dans le paramètre `prefix`, par exemple, `prefix=dir1/`.

Noms de clés

Pour les compartiments de répertoires, les sous-répertoires communs à plusieurs clés d'objet sont créés avec la première clé d'objet. Les clés d'objet supplémentaires pour le même sous-répertoire utilisent le sous-répertoire créé précédemment. Ce modèle vous permet de choisir les clés d'objet les mieux adaptées à l'application, avec une prise en charge égale des répertoires fragmentés et denses.

Gestion des accès

Les compartiments de répertoires ont tous les paramètres de blocage de l'accès public S3 activés par défaut au niveau des compartiments. La propriété des objets S3 est définie sur Propriétaire du

compartiment appliqué et les listes de contrôle d’accès (ACL) sont désactivées. Ces paramètres ne peuvent pas être modifiés.

Par défaut, les utilisateurs ne disposent pas d’autorisations pour les compartiments de répertoires et les opérations S3 Express One Zone. Pour accorder des autorisations d’accès pour les compartiments de répertoires, vous pouvez utiliser IAM pour créer des utilisateurs, des groupes ou des rôles, et attacher des autorisations à ces identités. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) for S3 Express One Zone.

Utilisation des compartiments de répertoires

Pour plus d’informations sur l’utilisation des compartiments de répertoires, consultez les rubriques suivantes.

Rubriques

- Règles de dénomination des compartiments de répertoires
- Création d’un compartiment de répertoires
- Affichage des propriétés d’un compartiment de répertoires
- Gestion des politiques de compartiment pour les compartiments de répertoires
- Vidage d’un compartiment de répertoires
- Suppression d’un compartiment de répertoires
- Établissement de la liste des compartiments de répertoires
- Utilisation HeadBucket avec des buckets de répertoires

Règles de dénomination des compartiments de répertoires

Lorsque vous créez un compartiment de répertoires dans Amazon S3, les règles de dénomination suivantes s’appliquent. Pour connaître les règles de dénomination des compartiments à usage général, consultez Règles de dénomination de compartiment.

Le nom d'un bucket d'annuaire se compose d'un nom de base que vous fournissez et d'un suffixe contenant l'ID de la zone de AWS disponibilité dans laquelle se trouve votre bucket. `--x-s3`

```
base-name--azid--x-s3
```

Par exemple, le nom de compartiment de répertoires suivant contient l’ID de zone de disponibilité `usw2-az1` :

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Lorsque vous créez un bucket de répertoire à l'aide de la console, un suffixe est automatiquement ajouté au nom de base que vous fournissez. Ce suffixe inclut l'ID de la zone de disponibilité que vous avez choisie.
> Lorsque vous créez un bucket d'annuaire à l'aide d'une API, vous devez fournir le suffixe complet, y compris l'ID de zone de disponibilité, dans votre demande. Pour obtenir la liste des ID de zone de disponibilité, consultezRégions et zones de disponibilité S3 Express One Zone.

Le nom d'un compartiment de répertoires :

- Soyez unique au sein de la zone Région AWS de disponibilité choisie.
- Le nom doit comporter entre 3 (min) et 63 (max) caractères, suffixe compris.
- Doit être composé uniquement de lettres minuscules, de chiffres et de traits d'union (-).
- Commencer et se terminer par une lettre ou un chiffre.
- Doit inclure le suffixe suivant :`--azid--x-s3`.

Création d'un compartiment de répertoires

Pour commencer à utiliser la classe de stockage Amazon S3 Express One Zone, vous devez créer un compartiment de répertoires. La classe de stockage S3 Express One Zone peut être utilisée uniquement avec des compartiments de répertoires. Elle prend en charge les cas d'utilisation à faible latence et accélère le traitement des données au sein d'une même zone de disponibilité. Si votre application est sensible aux performances et peut bénéficier d'une latence inférieure à dix millisecondes pour les demandes PUT et GET, nous vous recommandons de créer un compartiment de répertoires afin de pouvoir utiliser la classe de stockage S3 Express One Zone.

Il existe deux types de compartiments Amazon S3 : les compartiments à usage général et les compartiments de répertoires. Vous devez choisir le type de compartiment qui correspond le mieux à votre application et à vos exigences de performances. Les compartiments à usage général constituent le type de compartiment S3 d'origine. Les compartiments à usage général sont recommandés pour la plupart des cas d'utilisation et des modèles d'accès et autorisent le stockage

d'objets dans toutes les classes de stockage, à l'exception de S3 Express One Zone. Pour plus d'informations sur les compartiments à usage général, consultez Présentation des compartiments.

Les compartiments de répertoires utilisent la classe de stockage S3 Express One Zone, conçue pour être utilisée pour les charges de travail ou les applications critiques en termes de performances qui nécessitent une latence constante inférieure à dix millisecondes. S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d'objets et les ressources de calcul, ce qui assure la vitesse d'accès la plus élevée possible. Lorsque vous créez un bucket d'annuaire, vous pouvez éventuellement spécifier une zone de disponibilité Région AWS et une zone de disponibilité locales pour vos instances de calcul Amazon EC2, Amazon Elastic Kubernetes Service ou Amazon Elastic Container Service (Amazon ECS) afin d'optimiser les performances.

Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenu par l'accord de niveau de service Amazon S3. Pour plus d'informations, consultez Zone de disponibilité unique.

Les compartiments d'annuaire organisent les données de manière hiérarchique dans des répertoires, contrairement à la structure de stockage plate des compartiments à usage général. Il n'existe pas de limite de préfixe pour les compartiments de répertoires et les répertoires individuels peuvent faire l'objet d'une mise à l'échelle horizontale.

Pour plus d'informations sur les compartiments de répertoires, consultez Compartiments de répertoire.

Noms des compartiments de répertoires

Les noms des compartiments de répertoires doivent suivre ce format et respecter les règles de dénomination des compartiments de répertoires :

```
bucket-base-name--azid--x-s3
```

Par exemple, le nom de compartiment de répertoires suivant contient l'ID de zone de disponibilité `usw2-az1` :

```
bucket-base-name--usw2-az1--x-s3
```

Pour plus d'informations sur les règles de dénomination des compartiments de répertoires, consultez Règles de dénomination des compartiments de répertoires.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer un bucket.

   > **Note**
   >
   > Pour limiter la latence et les coûts, et répondre aux exigences légales, choisissez une région proche de vous. Les objets stockés dans une Région ne la quittent jamais, sauf si vous les transférez explicitement vers une autre Région. Pour obtenir la liste d'Amazon S3 Régions AWS, consultez la section sur les Service AWS points de terminaison dans le Référence générale d'Amazon Web Services.

3. Dans le panneau de navigation de gauche, choisissez Compartiments.
4. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
5. Sous Configuration générale, consultez l' Région AWS endroit où votre bucket sera créé.
6. Sous Type de compartiment, sélectionnez Répertoire.

   > **Note**
   >
   > - Si vous avez choisi une région qui ne prend pas en charge les compartiments de répertoire, l'option de type de compartiment disparaît et le type de compartiment est défini par défaut sur un compartiment à usage général. Pour créer un bucket d'annuaire, vous devez choisir une région prise en charge. Pour obtenir la liste des régions qui prennent en charge les compartiments d'annuaire et la classe de stockage Amazon S3 Express One Zone, consultezthe section called “Régions et zones de disponibilité S3 Express One Zone”.
   > - Une fois que vous avez créé le compartiment, vous ne pouvez pas modifier son type.

   Pour Zone de disponibilité, choisissez une zone de disponibilité locale pour vos services de calcul. Pour obtenir la liste des zones de disponibilité qui prennent en charge les compartiments

d'annuaire et la classe de stockage S3 Express One Zone, consultezthe section called “Régions et zones de disponibilité S3 Express One Zone”.

> **Note**
>
> La zone de disponibilité ne peut pas être modifiée une fois le compartiment créé.

7. Sous Zone de disponibilité, cochez la case pour reconnaître qu’en cas de panne de la zone de disponibilité, vos données pourraient être indisponibles ou perdues.

> **Important**
>
> Bien que les compartiments d'annuaire soient stockés sur plusieurs appareils au sein d'une même zone de disponibilité, les compartiments d'annuaire ne stockent pas les données de manière redondante entre les zones de disponibilité.

8. Pour Nom du compartiment, entrez un nom pour votre compartiment de répertoires.

   Le nom d’un compartiment de répertoires :

   - Soyez unique au sein de la zone Région AWS de disponibilité choisie.
   - Le nom doit comporter entre 3 (min) et 63 (max) caractères, suffixe compris.
   - Doit être composé uniquement de lettres minuscules, de chiffres et de traits d’union (-).
   - Commencer et se terminer par une lettre ou un chiffre.
   - Doit inclure le suffixe suivant :`--azid--x-s3`.

   Un suffixe est automatiquement ajouté au nom de base que vous fournissez lorsque vous créez un bucket de répertoire à l'aide de la console. Ce suffixe inclut l’ID de la zone de disponibilité que vous avez choisie.

   Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour plus d'informations sur l'attribution de noms à des compartiments, consultez Règles de dénomination de compartiment.

> **Important**
>
> N'incluez pas d'informations sensibles, telles que les numéros de compte, dans le nom du bucket. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

9. Sous Propriété de l'objet, le paramètre imposé au propriétaire du compartiment est automatiquement activé et toutes les listes de contrôle d'accès (ACL) sont désactivées. Pour les compartiments de répertoires, les listes ACL ne peuvent pas être activées.

   Listes ACL désactivées

   - Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations d'accès aux données du compartiment S3. Le compartiment utilise des stratégies exclusivement pour définir le contrôle des accès.

     La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

10. Dans les paramètres de blocage de l'accès public pour ce compartiment, tous les paramètres de blocage de l'accès public de votre compartiment de répertoire sont automatiquement activés. Ces paramètres ne peuvent pas être modifiés pour les compartiments de répertoire. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.
11. Dans les paramètres de chiffrement côté serveur, Amazon S3 applique le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) comme niveau de chiffrement de base pour tous les compartiments S3. Tous les téléchargements d'objets vers des compartiments de répertoire sont chiffrés avec SSE-S3. Pour les compartiments de répertoire, le type de chiffrement ne peut pas être modifié. Pour en savoir plus sur SSE-S3, consultez the section called “Clés de chiffrement gérées par Amazon S3 (SSE-S3)”.
12. Choisissez Créer un compartiment.

    Après avoir créé le compartiment, vous pouvez y ajouter des fichiers et des dossiers. Pour plus d'informations, consultez the section called “Utilisation d'objets dans un compartiment de répertoires”.

Utilisation des AWS kits de développement logiciel

SDK for Go

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for Go.

Example

```
var bucket = "..."

func runCreateBucket(c *s3.Client) {
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{
        Bucket: &bucket,
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            Location: &types.LocationInfo{
                Name: aws.String("usw2-az1"),
                Type: types.LocationTypeAvailabilityZone,
            },
            Bucket: &types.BucketInfo{
                DataRedundancy: types.DataRedundancySingleAvailabilityZone,
                Type:           types.BucketTypeDirectory,
            },
        },
    })
    var terr *types.BucketAlreadyOwnedByYou
    if errors.As(err, &terr) {
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))
        fmt.Printf("noop...\n")
        return
    }
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))
}
```

SDK for Java 2.x

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for Java 2.x.

Example

```
public static void createBucket(S3Client s3Client, String bucketName) {
```

```
   //Bucket name format is {base-bucket-name}--{az-id}--x-s3
   //example: doc-example-bucket--usw2-az1--x-s3 is a valid name for a directory
bucket created in
   //Region us-west-2, Availability Zone 2

   CreateBucketConfiguration bucketConfiguration =
CreateBucketConfiguration.builder()
            .location(LocationInfo.builder()
                    .type(LocationType.AVAILABILITY_ZONE)
                    .name("usw2-az1").build()) //this must match the Region and
Availability Zone in your bucket name
            .bucket(BucketInfo.builder()
                   .type(BucketType.DIRECTORY)
                   .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
                   .build()).build();
   try {

            CreateBucketRequest bucketRequest =
CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfigurat
            CreateBucketResponse response = s3Client.createBucket(bucketRequest);
            System.out.println(response);
   }

   catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
        }
   }
```

AWS SDK for JavaScript

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher
// To use with the preview build, place this in a folder
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/

import { S3 } from "@aws-sdk/client-s3";
```

```
const region = "us-east-1";
const zone = "use1-az4";
const suffix = `${zone}--x-s3`;

const s3 = new S3({ region });

const bucketName = `...--${suffix}`;

const createResponse = await s3.createBucket(
    { Bucket: bucketName,
      CreateBucketConfiguration: {Location: {Type: "AvailabilityZone", Name: zone},
      Bucket: { Type: "Directory", DataRedundancy: "SingleAvailabilityZone" }}
    }
   );
```

AWS SDK for .NET

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())
{
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest
    {

       BucketName = "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
       PutBucketConfiguration = new PutBucketConfiguration
       {
         BucketInfo = new BucketInfo { DataRedundancy =
 DataRedundancy.SingleAvailabilityZone, Type = BucketType.Directory },
         Location = new LocationInfo { Name = "usw2-az1", Type =
 LocationType.AvailabilityZone }
       }
     }).ConfigureAwait(false);
}
```

SDK for PHP

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for PHP.

Example

```
require 'vendor/autoload.php';
```

```
$s3Client = new S3Client([

    'region'      => 'us-east-1',
]);


$result = $s3Client->createBucket([
    'Bucket' => 'doc-example-bucket--use1-az4--x-s3',
    'CreateBucketConfiguration' => [
        'Location' => ['Name'=> 'use1-az4', 'Type'=> 'AvailabilityZone'],
        'Bucket' => ["DataRedundancy" => "SingleAvailabilityZone" ,"Type" =>
 "Directory"]   ],
]);
```

SDK for Python

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for Python (Boto3).

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def create_bucket(s3_client, bucket_name, availability_zone):
    '''
    Create a directory bucket in a specified Availability Zone

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to create; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :param availability_zone: String; Availability Zone ID to create the bucket in,
 for example, 'usw2-az1'
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
                'Location': {
                    'Type': 'AvailabilityZone',
                    'Name': availability_zone
                },
```

```
                'Bucket': {
                    'Type': 'Directory',
                    'DataRedundancy': 'SingleAvailabilityZone'
                }
            }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    availability_zone = 'usw2-az1'
    s3_client = boto3.client('s3', region_name = region)
    create_bucket(s3_client, bucket_name, availability_zone)
```

SDK for Ruby

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS SDK for Ruby.

Example

```
s3 = Aws::S3::Client.new(region:'us-west-2')
s3.create_bucket(
  bucket: "bucket_base_name--az_id--x-s3",
  create_bucket_configuration: {
    location: { name: 'usw2-az1', type: 'AvailabilityZone' },
    bucket: { data_redundancy: 'SingleAvailabilityZone', type: 'Directory' }
  }
)
```

À l'aide du AWS CLI

Cet exemple montre comment créer un bucket de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

Lorsque vous créez un bucket de répertoire, vous devez fournir les détails de configuration et utiliser la convention de dénomination suivante : *bucket-base-name*--*azid*--x-s3

```
aws s3api create-bucket
--bucket bucket-base-name--azid--x-s3
--create-bucket-configuration 'Location={Type=AvailabilityZone,Name=usw2-
az1},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory}'
--region us-west-2
```

Pour plus d'informations, consultez la section create-bucket dans le. AWS Command Line Interface

Affichage des propriétés d’un compartiment de répertoires

Vous pouvez consulter et configurer les propriétés d'un compartiment d'annuaire Amazon S3 à l'aide de la console Amazon S3. Pour plus d’informations, consultez Compartiments de répertoire et Qu’est-ce que S3 Express One Zone ?.

Utilisation de la console S3

1. Connectez-vous à la AWS Management Console et ouvrez la console Simple Storage Service (Amazon S3) à la page https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l’onglet Compartiments de répertoires.
4. Dans la liste Compartiments de répertoires, choisissez le nom du compartiment dont vous souhaitez afficher les propriétés.
5. Choisissez l’onglet Propriétés.
6. Dans l'onglet Propriétés, vous pouvez consulter les propriétés suivantes du bucket :

   - Vue d'ensemble du compartiment de répertoire : vous pouvez voir la zone de disponibilitéRégion AWS, le nom de ressource Amazon (ARN) et la date de création du compartiment.
   - Chiffrement par défaut : Amazon S3 applique un chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base de chiffrement pour tous les compartiments S3. Pour les compartiments de répertoires, ce paramètre ne peut pas être modifié. Amazon S3 chiffre un objet avant de l'enregistrer sur un disque et déchiffre l'objet quand vous le téléchargez. Pour plus d’informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Pour plus d’informations sur les fonctionnalités prises en charge pour les compartiments de répertoires, consultez Fonctionnalités de S3 Express One Zone.

Gestion des politiques de compartiment pour les compartiments de répertoires

Vous pouvez ajouter, supprimer, mettre à jour et consulter les politiques de compartiment pour les compartiments d'annuaire Amazon S3 à l'aide de la console Amazon S3 et des AWS kits SDK. Pour plus d’informations, consultez les rubriques suivantes. Pour plus d'informations sur les actions AWS Identity and Access Management (IAM) et les clés de condition prises en charge pour S3 Express One Zone, consultezAWS Identity and Access Management (IAM) pour S3 Express One Zone. Pour voir des exemples de politiques de compartiment pour les compartiments de répertoires, consultez Exemples de politiques de compartiment de répertoires pour S3 Express One Zone.

Rubriques

- Ajout d'une stratégie de compartiment
- Affichage d'une politique de compartiment
- Suppression d'une politique de compartiment

Ajout d'une stratégie de compartiment

Pour ajouter une politique de compartiment à un compartiment de répertoire, vous pouvez utiliser la console Amazon S3, AWS les SDK ou le AWS CLI.

Utilisation de la console S3

Pour créer ou modifier une stratégie de compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l’onglet Compartiments de répertoires.
4. Dans la liste Compartiments de répertoires, choisissez le nom du compartiment dans lequel vous souhaitez charger vos dossiers ou fichiers.
5. Choisissez l’onglet Permissions (Autorisations).

6. Sous Politique de compartiment, choisissez Modifier. La page Edit bucket policy (Modifier la politique de compartiment) s'affiche.
7. Pour générer une politique automatiquement, choisissez Générateur de politiques.

   Si vous choisissez le générateur de AWS politiques, celui-ci s'ouvre dans une nouvelle fenêtre.

   Si vous ne souhaitez pas utiliser le générateur de AWS politiques, vous pouvez ajouter ou modifier des instructions JSON dans la section Politique.

   a. Sur la page AWS Policy Generator (Générateur de politiques), pour Select Type of Policy (Sélectionner le type de politique), sélectionnez S3 Bucket Policy (Politique de compartiment S3).
   b. Ajoutez une instruction en saisissant les informations dans les champs fournis, puis choisissez Add Statement (Ajouter une instruction). Répétez cette étape pour tous les énoncés que vous souhaitez ajouter. Pour plus d'informations sur ces champs, consultez la Référence des éléments de stratégie IAM JSON dans le Guide de l'utilisateur IAM.

      > **Note**
      >
      > Pour vous faciliter la tâche, la page Modifier la politique du bucket affiche l'ARN du bucket (Amazon Resource Name) du bucket actuel au-dessus du champ de texte de la politique. Vous pouvez copier cet ARN pour l'utiliser dans les instructions de la page AWS Policy Generator (Générateur de politique).

   c. Une fois que vous avez fini d'ajouter des instructions, choisissez Generate Policy (Générer une stratégie).
   d. Copiez le texte de stratégie généré, choisissez Close (Fermer) et revenez à la page Edit bucket policy (Modifier la stratégie de compartiment) dans la console Amazon S3.
8. Dans la zone Politique, modifiez la politique existante ou collez la politique de compartiment depuis le générateur de AWS politiques. Veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions avant d'enregistrer votre stratégie.

   > **Note**
   >
   > Les stratégies de compartiment sont limitées à une taille de 20 Ko.

9. Choisissez Save changes (Enregistrer les modifications), ce qui vous ramène à l'onglet Permissions (Autorisations).

Utilisation des AWS SDK

SDK for Java 2.x

Example

PutBucketPolicy AWS SDK for Java 2.x

```
public static void setBucketPolicy(S3Client s3Client, String bucketName, String
 policyText) {

      //sample policy text
      /**
       * policy_statement = {
        *         'Version': '2012-10-17',
        *         'Statement': [
        *             {
        *                 'Sid': 'AdminPolicy',
        *                 'Effect': 'Allow',
        *                 'Principal': {
        *                     "AWS": "111122223333"
        *                 },
        *                 'Action': 's3express:*',
        *                 'Resource':
 'arn:aws:s3express:region:111122223333:bucket/bucket-base-name--azid--x-s3'
        *             }
        *         ]
        *     }
        */
        System.out.println("Setting policy:");
        System.out.println("----");
        System.out.println(policyText);
        System.out.println("----");
        System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

        try {
            PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
                    .bucket(bucketName)
                    .policy(policyText)
                    .build();
            s3Client.putBucketPolicy(policyReq);
            System.out.println("Done!");
        }
```

```
        catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

En utilisant le AWS CLI

Cet exemple montre comment ajouter une politique de compartiment à un compartiment de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api put-bucket-policy --bucket bucket-base-name--azid--x-s3 --policy file://
bucket_policy.json
```

bucket_policy.json :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AdminPolicy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express*",
            "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/"
        }
    ]
}
```

Pour plus d'informations, consultez put-bucket-policyle AWS Command Line Interface.

Affichage d'une politique de compartiment

Pour consulter une politique de compartiment pour un compartiment de répertoire, utilisez les exemples suivants.

En utilisant le AWS CLI

Cet exemple montre comment afficher la politique de compartiment attachée à un compartiment de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api get-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Pour plus d'informations, consultez get-bucket-policyle AWS Command Line Interface.

Suppression d'une politique de compartiment

Pour supprimer une politique de compartiment pour un compartiment de répertoire, utilisez les exemples suivants.

Utilisation des AWS SDK

SDK for Java 2.x

Example

`DeleteBucketPolicy` AWS SDK for Java 2.x

```
public static void deleteBucketPolicy(S3Client s3Client, String bucketName) {
      try {
          DeleteBucketPolicyRequest deleteBucketPolicyRequest =
 DeleteBucketPolicyRequest
                  .builder()
                  .bucket(bucketName)
                  .build()
          s3Client.deleteBucketPolicy(deleteBucketPolicyRequest);
          System.out.println("Successfully deleted bucket policy");
      }

      catch (S3Exception e) {
          System.err.println(e.awsErrorDetails().errorMessage());
          System.exit(1);
      }
```

En utilisant le AWS CLI

Cet exemple montre comment supprimer une politique de compartiment pour un compartiment de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--azid--x-s3
```

Pour plus d'informations, consultez delete-bucket-policyle AWS Command Line Interface.

Vidage d’un compartiment de répertoires

Vous pouvez vider un compartiment d'annuaire Amazon S3 à l'aide de la console Amazon S3. Pour plus d’informations sur les compartiments de répertoires, consultez Compartiments de répertoire.

Avant de vider un compartiment de répertoires, notez les points suivants :

- Lorsque vous videz un compartiment de répertoires, vous supprimez tous les objets, mais vous conservez le compartiment de répertoires.
- Une fois que vous avez vidé un compartiment de répertoire, l'action vide ne peut pas être annulée.
- Les objets ajoutés au compartiment de répertoire alors que l'action du compartiment vide est en cours d'exécution peuvent être supprimés.

Si vous souhaitez également supprimer le bucket, notez ce qui suit :

- Tous les objets figurant dans le compartiment de répertoires doivent être supprimés avant que le compartiment lui-même puisse être supprimé.
- Les chargements partitionnés en cours dans le compartiment de répertoires doivent être abandonnés avant que le compartiment lui-même puisse être supprimé.

> **Note**
>
> La `s3 rm` commande via la AWS Command Line Interface (CLI), l'`delete`opération via Mountpoint et le bouton d'option Empty bucket via le ne permettent pas de supprimer les téléchargements AWS Management Console partitionnés en cours dans un compartiment de répertoire. Pour supprimer ces téléchargements partitionnés en cours, utilisez l'`ListMultipartUploads`opération pour répertorier les téléchargements partitionnés

en cours dans le bucket et utilisez l'`AbortMultupartUpload`opération pour abandonner tous les téléchargements partitionnés en cours.

Pour supprimer un compartiment de répertoires, consultez Suppression d’un compartiment de répertoires. Pour annuler un téléchargement partitionné en cours, consultez. the section called “Interruption d'un chargement partitionné”

Pour vider un compartiment à usage général, consultez Vider un compartiment.

Utilisation de la console S3

Pour vider un compartiment de répertoire

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l’onglet Compartiments de répertoires.
4. Cliquez sur le bouton d'option à côté du nom du compartiment que vous souhaitez vider, puis choisissez Empty.
5. Dans la page Empty bucket (Vider le compartiment), confirmez que vous souhaitez vider le compartiment en saisissant **`permanently delete`** dans le champ de texte, puis choisissez Empty (Vider).
6. Surveillez la progression du processus de vidange du compartiment sur la page Compartiment vide : état.

Suppression d’un compartiment de répertoires

Vous ne pouvez supprimer que des compartiments d'annuaire Amazon S3 vides. Avant de supprimer votre compartiment de répertoire, vous devez supprimer tous les objets qu'il contient et abandonner tous les téléchargements partitionnés en cours.

Pour vider un compartiment de répertoires, consultez Vidage d’un compartiment de répertoires. Pour annuler un téléchargement partitionné en cours, consultez. the section called “Interruption d'un chargement partitionné”

Pour supprimer un compartiment à usage général, consultez Suppression d'un compartiment.

Utilisation de la console S3

Après avoir vidé votre compartiment de répertoire et abandonné tous les téléchargements partitionnés en cours, vous pouvez le supprimer.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l'onglet Compartiments de répertoires.
4. Dans la liste des compartiments du répertoire, cliquez sur le bouton d'option situé à côté du compartiment que vous souhaitez supprimer.
5. Sélectionnez Delete (Supprimer).
6. Sur la page Supprimer le compartiment, entrez le nom du compartiment dans le champ de texte pour confirmer la suppression de votre compartiment.

   > **Important**
   >
   > La suppression d'un compartiment de répertoires ne peut pas être annulée.

7. Pour supprimer votre compartiment de répertoires, choisissez Supprimer le compartiment.

Utilisation des AWS kits de développement logiciel

Les exemples suivants suppriment un bucket de répertoire à l'aide des touches AWS SDK for Java 2.x et AWS SDK for Python (Boto3).

SDK for Java 2.x

Example

```
public static void deleteBucket(S3Client s3Client, String bucketName) {

    try {
        DeleteBucketRequest del = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();
        s3Client.deleteBucket(del);
        System.out.println("Bucket " + bucketName + " has been deleted");
    }
```

```
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_bucket(s3_client, bucket_name):
    '''
    Delete a directory bucket in a specified Region

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to delete; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :return: True if bucket is deleted, else False
    '''

    try:
        s3_client.delete_bucket(Bucket = bucket_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
```

À l'aide du AWS CLI

Cet exemple montre comment supprimer un bucket de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api delete-bucket --bucket bucket-base-name--azid--x-s3 --region us-west-2
```

Pour plus d'informations, consultez la section delete-bucket dans le. AWS Command Line Interface

Établissement de la liste des compartiments de répertoires

Les exemples suivants montrent comment répertorier les compartiments de répertoires à l'aide des AWS SDK et de la CLI AWS .

Utilisation des AWS SDK

SDK for Java 2.x

Example

L'exemple suivant répertorie les compartiments de répertoire à l'aide du AWS SDK for Java 2.x.

```
public static void listBuckets(S3Client s3Client) {
       try {
           ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
           ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
           if (response.hasBuckets()) {
               for (Bucket bucket: response.buckets()) {
                   System.out.println(bucket.name());
                   System.out.println(bucket.creationDate());
               }
           }
       }

       catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

SDK for Python

Example

L'exemple suivant répertorie les compartiments de répertoire à l'aide du AWS SDK for Python (Boto3).

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
'''
Prints a list of all directory buckets in a Region

:param s3_client: boto3 S3 client
:return: True if there are buckets in the Region, else False
'''
try:
    response = s3_client.list_directory_buckets()
    for bucket in response['Buckets']:
        print (bucket['Name'])
except ClientError as e:
    logging.error(e)
    return False
return True


if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
    list_directory_buckets(s3_client)
```

AWS SDK for .NET

Example

L'exemple suivant répertorie les compartiments de répertoire à l'aide du AWS SDK for .NET.

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
 ListDirectoryBucketsRequest
```

```
{
  MaxDirectoryBuckets = 10
  }).ConfigureAwait(false);
```

SDK for PHP

Example

L'exemple suivant répertorie les compartiments de répertoire à l'aide du AWS SDK for PHP.

```
require 'vendor/autoload.php';

$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

L'exemple suivant répertorie les compartiments de répertoire à l'aide du AWS SDK for Ruby.

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

En utilisant le AWS CLI

L'`list-directory-buckets`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour répertorier vos compartiments de répertoire dans la région *`us-east-1`*. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api list-directory-buckets --region us-east-1
```

Pour plus d'informations, consultez la section list-directory-buckets dans la référence des commandes AWS CLI .

Utilisation **`HeadBucket`** avec des buckets de répertoires

Les exemples de AWS SDK suivants montrent comment utiliser l'opération d'`HeadBucket`API pour déterminer si un compartiment d'annuaire Amazon S3 existe et si vous êtes autorisé à y accéder.

Utilisation des AWS SDK

L' AWS SDK for Java 2.x exemple suivant montre comment déterminer si un bucket existe et si vous êtes autorisé à y accéder.

SDK for Java 2.x

Example

AWS SDK for Java 2.x

```
public static void headBucket(S3Client s3Client, String bucketName) {
   try {
        HeadBucketRequest headBucketRequest = HeadBucketRequest
                .builder()
                .bucket(bucketName)
                .build();
        s3Client.headBucket(headBucketRequest);
        System.out.format("Amazon S3 bucket: \"%s\" found.", bucketName);
   }

   catch (S3Exception e) {
       System.err.println(e.awsErrorDetails().errorMessage());
       System.exit(1);
   }
}
```

En utilisant le AWS CLI

L'`head-bucket`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour déterminer si un bucket de répertoire existe et si vous êtes autorisé à y accéder. Pour exécuter cette commande, remplacez les espaces réservés saisis par l'utilisateur par vos propres informations.

```
aws s3api head-bucket --bucket bucket-base-name--azid--x-s3
```

Pour plus d’informations, consultez la section head-bucket dans la référence des commandes AWS CLI .

Utilisation d'objets dans un bucket de répertoire

Après avoir créé un compartiment d'annuaire Amazon S3, vous pouvez travailler avec des objets à l'aide de la console Amazon S3 AWS Command Line Interface (AWS CLI) et AWS des kits SDK.

Pour plus d'informations sur les opérations d'objets en masse avec des objets stockés dans la classe de stockage S3 Express One Zone, consultezGestion des objets. Pour plus d'informations sur l'importation, le chargement, la copie, la suppression et le téléchargement d'objets, ainsi que sur la lecture des métadonnées des objets contenus dans des compartiments de répertoire, consultez les rubriques suivantes.

Rubriques

- Importation d’objets dans un compartiment de répertoires
- Utilisation des opérations par lots avec S3 Express One Zone
- Chargement d’un objet dans un compartiment de répertoires
- Utilisation de téléchargements partitionnés avec des compartiments de répertoires
- Copie d’un objet vers un compartiment de répertoires
- Suppression d’un objet dans un compartiment de répertoires
- Téléchargement d'un objet dans un bucket de répertoire
- Utilisation HeadObject avec des buckets de répertoires

Importation d’objets dans un compartiment de répertoires

Après avoir créé un compartiment de répertoires dans Amazon S3, vous pouvez remplir le nouveau compartiment avec des données à l’aide de l’action d’importation. L’importation est une méthode simplifiée pour créer des tâches d’opérations par lots S3 afin de copier des objets depuis des compartiments à usage général vers des compartiments de répertoires.

> **Note**
>
> Les limites suivantes s’appliquent aux tâches d’importation :

- Le compartiment source et le compartiment de destination doivent se trouver dans la même Région AWS et dans le même compte.
- Le compartiment source ne peut pas être un compartiment de répertoires.
- Les objets de plus de 5 Go ne sont pas pris en charge et seront omis dans l'opération de copie.
- Les objets des classes de stockage de niveau Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access et Intelligent-Tiering Deep Archive doivent être restaurés avant de pouvoir être importés.
- Les objets importés avec des algorithmes de somme de contrôle MD5 sont convertis pour utiliser les sommes de contrôle CRC32.
- Les objets importés utilisent le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
- Les objets importés utilisent la classe de stockage Express One Zone, dont la structure tarifaire est différente de celle des classes de stockage utilisées par les buckets à usage général. Tenez compte de cette différence de coût lorsque vous importez un grand nombre d'objets.

Lorsque vous configurez une tâche d'importation, vous spécifiez le compartiment source ou le préfixe source à partir duquel les objets existants seront copiés. Vous fournissez également un rôle AWS Identity and Access Management (IAM) autorisé à accéder aux objets sources. Amazon S3 lance ensuite une tâche d'opérations par lots qui copie les objets et applique automatiquement les paramètres de classe de stockage et de somme de contrôle appropriés.

Pour configurer les tâches d'importation, vous utilisez la console Amazon S3.

Utilisation de la console Amazon S3

Pour importer des objets dans un compartiment de répertoires

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Compartiments, puis l'onglet Compartiments de répertoires. Choisissez le bouton d'option en regard du compartiment de répertoires dans lequel vous souhaitez importer des objets.
3. Choisissez Import (Importer).

4. Pour Source, entrez le compartiment à usage général (ou le chemin du compartiment, préfixe inclus) qui contient les objets que vous souhaitez importer. Pour choisir un compartiment à usage général existant dans une liste, choisissez Parcourir S3.
5. Pour Autorisation d'accéder aux objets sources et de les copier, effectuez l'une des opérations suivantes pour spécifier un rôle IAM doté des autorisations nécessaires pour importer vos objets sources :

   - Pour autoriser Amazon S3 à créer un nouveau rôle IAM en votre nom, choisissez Créer un nouveau rôle IAM.

     > **Note**
     >
     > Si vos objets sources sont chiffrés avec un chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), ne choisissez pas l'option Créer un nouveau rôle IAM. Spécifiez plutôt un rôle IAM existant disposant de l'autorisation `kms:Decrypt`.
     > Amazon S3 utilisera cette autorisation pour déchiffrer vos objets. Pendant le processus d'importation, Amazon S3 rechiffrera ensuite ces objets à l'aide d'un chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3).

   - Pour choisir un rôle IAM existant dans une liste, choisissez Sélectionner parmi les rôles IAM existants.
   - Pour spécifier un rôle IAM existant en entrant son Amazon Resource Name (ARN), choisissez Saisir l'ARN du rôle IAM, puis entrez l'ARN dans le champ correspondant.
6. Passez en revue les informations affichées dans les sections Destination et Paramètres de l'objet copié. Si les informations de la section Destination sont correctes, choisissez Importer pour démarrer la tâche de copie.

   La console Amazon S3 affiche le statut de votre nouvelle tâche sur la page Opérations par lot. Pour plus d'informations sur la tâche, choisissez le bouton d'option à côté du nom de la tâche, puis dans le menu Actions, choisissez Afficher les détails. Pour ouvrir le compartiment de répertoires dans lequel les objets seront importés, choisissez Afficher la destination de l'importation.

Utilisation des opérations par lots avec S3 Express One Zone

Vous pouvez utiliser les opérations par lots Amazon S3 pour effectuer des opérations sur des objets stockés dans des compartiments S3. Pour en savoir plus sur les opérations par lots S3, consultez Exécution des opérations par lots à grande échelle sur des objets Amazon S3.

Les rubriques suivantes traitent de l'exécution d'opérations par lots sur des objets stockés dans la classe de stockage S3 Express One Zone dans des compartiments de répertoire.

Rubriques

Utilisation des opérations par lots avec les compartiments de répertoires

Vous pouvez exécuter les opérations Copy et AWS LambdaInvoke sur des objets stockés dans des compartiments de répertoire. Avec Copy, vous pouvez copier des objets entre des compartiments du même type (par exemple, d'un compartiment de répertoire vers un compartiment de répertoire). Vous pouvez également copier entre des compartiments à usage général et des compartiments de répertoires. Avec la fonction AWS Lambda Invoke, vous pouvez utiliser une fonction Lambda pour effectuer des actions sur des objets figurant dans votre compartiment de répertoires avec du code que vous définissez.

Copie d'objets

Vous pouvez copier entre des compartiments de même type ou entre des compartiments de répertoires et des compartiments à usage général. Lorsque vous effectuez une copie dans un compartiment d'annuaire, vous devez utiliser le format Amazon Resource Name (ARN) correct pour ce type de compartiment. Le format d'ARN d'un compartiment de répertoires est `arn:aws:s3express:`*`region`*`:`*`account-id`*`:bucket/`*`bucket-base-name`*`--x-s3`.

Vous pouvez également remplir votre compartiment de répertoires avec des données à l'aide de l'action Importer de la console S3. L'action Importer est une méthode simplifiée pour créer des tâches d'opérations par lots afin de copier des objets depuis des compartiments à usage général vers des compartiments de répertoires. Pour les tâches de copie Importer depuis des compartiments à usage général vers des compartiments de répertoires, S3 génère automatiquement un manifeste. Pour plus d'informations, consultez les sections Importation d'objets dans un compartiment de répertoire et Spécification d'un manifeste.

Invocation de fonctions Lambda () **`LambdaInvoke`**

L'utilisation d'opérations par lots pour invoquer des fonctions Lambda agissant sur des compartiments de répertoires est soumise à des exigences particulières. Par exemple, vous devez structurer votre demande Lambda à l'aide d'un schéma d'invocation v2 JSON et spécifier à quel InvocationSchemaVersion 2.0 moment vous créez la tâche. Pour plus d'informations, consultez la section AWS LambdaFonction Invoke.

Principales différences

Voici une liste des principales différences lorsque vous utilisez Batch Operations pour effectuer des opérations groupées sur des objets stockés dans des compartiments de répertoire avec la classe de stockage S3 Express One Zone :

- Amazon S3 chiffre automatiquement tous les nouveaux objets chargés dans un compartiment S3. La configuration de chiffrement par défaut d'un compartiment S3 est toujours activée et est au minimum définie sur le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3). Pour les compartiments de répertoire, seul SSSE-S3 est pris en charge. Si vous faites une `CopyObject` demande qui définit le chiffrement côté serveur avec des clés fournies par le client (SSE-C) ou le chiffrement côté serveur avec des clés () AWS Key Management Service (SSE-KMSAWS KMS) sur un compartiment de répertoire (source ou destination), la réponse renvoie une erreur HTTP. `400 (Bad Request)`
- Les objets figurant dans des compartiments de répertoires ne peuvent pas être balisés. Vous ne pouvez spécifier qu'un jeu de balises vide. Par défaut, les opérations par lots copient les balises. Si vous copiez un objet comportant des balises entre des compartiments à usage général et des compartiments de répertoire, vous recevez une `501 (Not Implemented)` réponse.
- S3 Express One Zone vous offre la possibilité de choisir l'algorithme de somme de contrôle utilisé pour valider vos données lors des chargements ou des téléchargements. Vous pouvez sélectionner l'un des algorithmes de contrôle d'intégrité des données Secure Hash Algorithms (SHA) ou Cyclic Redundancy Check (CRC) suivants : CRC32, CRC32C, SHA-1 ou SHA-256. Les checksums basés sur MD5 ne sont pas pris en charge avec la classe de stockage S3 Express One Zone.
- Par défaut, tous les compartiments Amazon S3 définissent le paramètre S3 Object Ownership comme étant appliqué par le propriétaire du compartiment et les listes de contrôle d'accès (ACL) sont désactivées. Pour les compartiments de répertoires, ce paramètre ne peut pas être modifié. Vous pouvez copier un objet à partir de compartiments à usage général vers des compartiments de répertoires. Toutefois, vous ne pouvez pas remplacer l'ACL par défaut lorsque vous effectuez une copie vers ou depuis un bucket de répertoire.

- Quelle que soit la manière dont vous spécifiez votre manifeste, la liste elle-même doit être stockée dans un compartiment à usage général. Batch Operations ne peut pas importer de manifestes existants depuis (ni enregistrer les manifestes générés dans) des compartiments de répertoire. Toutefois, les objets décrits dans le manifeste peuvent être stockés dans des compartiments de répertoires.
- Batch Operations ne peut pas spécifier un compartiment de répertoire comme emplacement dans un rapport d'inventaire S3. Les rapports d'inventaire ne prennent pas en charge les compartiments de répertoires. Vous pouvez créer un fichier manifeste pour les objets d'un bucket de répertoire en utilisant l'opération `ListObjectsV2` API pour répertorier les objets. Vous pouvez ensuite insérer la liste dans un fichier CSV.

Octroi de l'accès à

Pour effectuer des tâches de copie, vous devez disposer des autorisations suivantes :

- Pour copier des objets d’un compartiment de répertoires vers un autre, vous devez disposer de l’autorisation `s3express:CreateSession`.
- Pour copier des objets à partir de compartiments de répertoires vers des compartiments à usage général, vous devez disposer de l’autorisation `s3express:CreateSession` et de l’autorisation `s3:PutObject` permettant d’écrire la copie des objets dans le compartiment de destination.
- Pour copier des objets depuis des compartiments à usage général vers des compartiments de répertoire, vous devez avoir l'`s3express:CreateSession`autorisation et l'`s3:GetObject`autorisation de lire l'objet source copié.

  Pour plus d'informations, veuillez consulter CopyObject dans la Référence d'API Amazon Simple Storage Service.
- Pour invoquer une fonction Lambda, vous devez accorder des autorisations à votre ressource sur la base de votre fonction Lambda. Pour déterminer les autorisations requises, vérifiez les opérations d'API correspondantes.

Chargement d’un objet dans un compartiment de répertoires

Après avoir créé un compartiment d'annuaire Amazon S3, vous pouvez y charger des objets. Les exemples suivants montrent comment télécharger un objet dans un compartiment de répertoire à l'aide de la console S3 et des AWS kits de développement logiciel. Pour plus d'informations sur les

opérations de téléchargement d'objets en masse avec S3 Express One Zone, consultezGestion des objets.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l'onglet Compartiments de répertoires.
4. Choisissez le nom du compartiment dans lequel vous souhaitez télécharger vos dossiers ou fichiers.
5. Dans la liste des objets, choisissez Upload.
6. Sur la page de téléchargement, effectuez l'une des opérations suivantes :
   - Faites glisser les fichiers et les dossiers vers la zone de téléchargement en pointillés.
   - Choisissez Ajouter des fichiers ou Ajouter un dossier, choisissez les fichiers ou les dossiers à télécharger, puis choisissez Ouvrir ou Charger.
7. Sous Checksum, choisissez la fonction Checksum que vous souhaitez utiliser.

   (Facultatif) Si vous téléchargez un seul objet d'une taille inférieure à 16 Mo, vous pouvez également spécifier une valeur de somme de contrôle précalculée. Lorsque vous fournissez une valeur précalculée, Amazon S3 la compare à la valeur qu'il calcule à l'aide de la fonction de somme de contrôle sélectionnée. Si les valeurs ne correspondent pas, le téléchargement ne démarrera pas.
8. Les options des sections Autorisations et Propriétés sont automatiquement définies selon les paramètres par défaut et ne peuvent pas être modifiées. Le blocage de l'accès public est automatiquement activé, et le versionnage S3 et le verrouillage des objets S3 ne peuvent pas être activés pour les compartiments de répertoire.

   (Facultatif) Si vous souhaitez ajouter des métadonnées sous forme de paires clé-valeur à vos objets, développez la section Propriétés, puis choisissez Ajouter des métadonnées dans la section Métadonnées.
9. Pour télécharger les fichiers et dossiers répertoriés, choisissez Upload.

   Amazon S3 charge les objets et les dossiers. Lorsque le chargement est terminé, un message de réussite s'affiche sur la page Charger : statut.

Utilisation des AWS kits de développement logiciel

SDK for Java 2.x

Example

```
public static void putObject(S3Client s3Client, String bucketName, String objectKey,
 Path filePath) {
       //Using File Path to avoid loading the whole file into memory
       try {
           PutObjectRequest putObj = PutObjectRequest.builder()
                   .bucket(bucketName)
                   .key(objectKey)
                   //.metadata(metadata)
                   .build();
           s3Client.putObject(putObj, filePath);
           System.out.println("Successfully placed " + objectKey +" into bucket
 "+bucketName);

       }

       catch (S3Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
}
```

SDK for Python

Example

```
import boto3
import botocore
from botocore.exceptions import ClientError


def put_object(s3_client, bucket_name, key_name, object_bytes):
    """
    Upload data to a directory bucket.
    :param s3_client: The boto3 S3 client
    :param bucket_name: The bucket that will contain the object
    :param key_name: The key of the object to be uploaded
    :param object_bytes: The data to upload
```

```
    """
    try:
        response = s3_client.put_object(Bucket=bucket_name, Key=key_name,
                                Body=object_bytes)
        print(f"Upload object '{key_name}' to bucket '{bucket_name}'.")
        return response
    except ClientError:
        print(f"Couldn't upload object '{key_name}' to bucket '{bucket_name}'.")
        raise

def main():
    # Share the client session with functions and objects to benefit from S3 Express
 One Zone auth key
    s3_client = boto3.client('s3')
    # Directory bucket name must end with --azid--x-s3
    resp = put_object(s3_client, 'doc-bucket-example--use1-az5--x-s3', 'sample.txt',
 b'Hello, World!')
    print(resp)

if __name__ == "__main__":
    main()
```

À l'aide du AWS CLI

L'`put-object`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour charger un objet depuis Amazon S3. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api put-object --bucket bucket-base-name--azid--x-s3 --key sampleinut/file001.bin
 --body bucket-seed/file001.bin
```

Pour plus d'informations, consultez la section put-object dans la référence des commandes AWS CLI .

Utilisation de téléchargements partitionnés avec des compartiments de répertoires

Vous pouvez utiliser le processus de téléchargement en plusieurs parties pour télécharger un seul objet sous la forme d'un ensemble de parties. Chaque partie est une portion contiguë des données de l'objet. Vous pouvez charger ces parties d'objet indépendamment et dans n'importe quel ordre. Si

le transfert d'une partie échoue, vous pouvez la retransférer sans affecter les autres. Une fois toutes les parties de l'objet chargées, Amazon S3 les assemble et crée l'objet. En général, lorsque l'objet atteint la taille de 100 Mo, vous devez préférer les chargements partitionnés au chargement d'objet en une seule opération.

L'utilisation du chargement partitionné offre les avantages suivants :

- Improved throughput (Meilleur débit) — vous pouvez charger des parties en parallèle pour améliorer le débit.
- Restauration rapide en cas de problème réseau : les pièces de petite taille minimisent l'impact du redémarrage d'un téléchargement échoué en raison d'une erreur réseau.
- Pause and resume object uploads (Interruption et reprise des chargements d'objet) — vous pouvez charger des parties d'objet au fil du temps. Une fois que vous avez lancé un téléchargement en plusieurs parties, il n'y a aucune date d'expiration. Vous devez explicitement terminer ou abandonner le téléchargement partitionné.
- Begin an upload before you know the final object size (Lancement d'un chargement avant de connaître la taille finale de l'objet) — vous pouvez charger un objet à mesure que vous le créez.

Nous vous recommandons d'utiliser les téléchargements partitionnés de la manière suivante :

- Si vous chargez des objets volumineux sur un réseau stable à large bande passante, utilisez les téléchargements partitionnés pour optimiser l'utilisation de la bande passante disponible en téléchargeant des parties d'objets en parallèle pour des performances multithread.
- Si vous effectuez un téléchargement sur un réseau irrégulier, utilisez les téléchargements partitionnés pour augmenter la résilience face aux erreurs réseau en évitant les redémarrages de téléchargement. Lorsque vous utilisez des téléchargements partitionnés, vous devez réessayer de télécharger uniquement les parties interrompues pendant le téléchargement. Vous n'avez pas besoin de redémarrer le chargement de vos objets depuis le début.

Lorsque vous utilisez des chargements partitionnés pour charger des objets vers la classe de stockage Amazon S3 Express One Zone dans des compartiments de répertoire, le processus de téléchargement partitionné est similaire au processus d'utilisation du téléchargement partitionné pour télécharger des objets dans des compartiments à usage général. Cependant, il existe quelques différences importantes.

Pour plus d'informations sur l'utilisation des téléchargements partitionnés pour télécharger des objets dans S3 Express One Zone, consultez les rubriques suivantes.

Rubriques

Le processus de téléchargement en plusieurs parties

Un téléchargement en plusieurs parties est un processus en trois étapes :

- Vous lancez le téléchargement.
- Vous chargez les parties de l'objet.
- Une fois que vous avez chargé toutes les parties, vous pouvez terminer le téléchargement en plusieurs parties.

Dès réception de la demande de téléchargement en plusieurs parties complète, Amazon S3 construit l'objet à partir des parties téléchargées, et vous pouvez ensuite accéder à l'objet comme vous le feriez pour n'importe quel autre objet de votre compartiment.

Lancement du chargement partitionné

Lorsque vous envoyez une demande pour lancer un chargement partitionné, Amazon S3 renvoie une réponse avec un ID de chargement, qui est un identifiant unique pour le chargement partitionné. Vous devez inclure cet ID de chargement dès que vous chargez les parties, listez les parties, terminez un chargement ou interrompez un chargement.

Chargement de parties

Lorsque vous chargez une partie, outre l'ID de chargement, vous devez spécifier un numéro de partie. Lorsque vous utilisez un téléchargement partitionné avec S3 Express One Zone, les numéros de pièce partitionnés doivent être des numéros de pièce consécutifs. Si vous essayez de traiter une demande de téléchargement en plusieurs parties avec des numéros de pièce non consécutifs, une erreur `HTTP 400 Bad Request` (commande de pièce non valide) est générée.

Un numéro de pièce identifie de manière unique une pièce et sa position dans l'objet que vous chargez. Si vous chargez une nouvelle pièce en utilisant le même numéro de pièce qu'une pièce précédemment téléchargée, la pièce précédemment téléchargée est remplacée.

Chaque fois que vous chargez une partie, Amazon S3 renvoie un en-tête de balise d'entité ETag dans sa réponse. Pour chaque chargement de partie, vous devez enregistrer le numéro de partie et la valeur ETag. Les valeurs ETag pour tous les chargements de parties d'objets resteront les mêmes, mais un numéro de pièce différent sera attribué à chaque partie. Vous devez inclure ces valeurs dans la demande ultérieure pour terminer le chargement partitionné.

Amazon S3 chiffre automatiquement tous les nouveaux objets chargés dans un compartiment S3. Dans le cadre d'un chargement partitionné, si vous ne spécifiez pas d'informations de chiffrement dans votre demande, le paramètre de chiffrement des parties chargées est défini sur la configuration de chiffrement par défaut du compartiment de destination. La configuration de chiffrement par défaut d'un compartiment Amazon S3 est toujours activée et est au minimum définie sur le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3). Pour les compartiments de répertoire, seul le SSE-S3 est pris en charge. Pour plus d'informations, consultez Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

Fin du chargement partitionné

Lorsque vous effectuez un chargement partitionné, Amazon S3 crée l'objet en concaténant les parties par ordre croissant en fonction du numéro de pièce. À l'issue d'une demande de chargement complet, les parties n'existent plus.

Votre demande complète de téléchargement en plusieurs parties doit inclure l'ID de téléchargement et une liste des deux numéros de pièce et des valeurs ETag correspondantes. La réponse d'Amazon S3 inclut une valeur ETag qui identifie de façon unique les données d'objet combinées. Cette balise d'entité n'est pas un hachage MD5 des données d'objet.

Listes de chargement partitionné

Vous pouvez lister les parties d'un chargement partitionné spécifique ou de tous les chargements partitionnés en cours. L'opération de liste des parties renvoie des informations sur les parties que vous avez chargées pour un chargement partitionné spécifique. Pour chaque demande de liste des parties, Amazon S3 renvoie des informations sur les parties pour le chargement partitionné spécifié, pour 1 000 parties maximum. Si le chargement partitionné compte plus de 1 000 parties, vous devez utiliser la pagination pour récupérer toutes les parties.

La liste des pièces renvoyée n'inclut pas les pièces dont le téléchargement n'est pas terminé. En utilisant l'opération d'affichage des chargements partitionnés, vous pouvez obtenir la liste des chargements partitionnés qui sont en cours.

Un chargement partitionné en cours est un chargement que vous avez lancé, mais que vous n'avez pas encore terminé ou interrompu. Chaque demande renvoie 1,000 chargements partitionnés maximum. S'il y a plus de 1 000 chargements partitionnés en cours, vous devez envoyer des demandes supplémentaires pour récupérer les chargements partitionnés restants. Utilisez la liste renvoyée uniquement pour la vérification. N'utilisez pas le résultat de la liste lorsque vous envoyez une requête de chargement partitionné complet. Au lieu de cela, conservez votre propre liste des numéros de parties que vous avez spécifiés lors du chargement des parties ainsi que les valeurs ETag correspondantes renvoyées par Amazon S3.

Pour plus d'informations sur les listes de téléchargement partitionné, consultez le ListPartsmanuel Amazon Simple Storage Service API Reference.

Totaux de contrôle avec les opérations de chargement partitionné

Lorsque vous chargez un objet, vous pouvez spécifier un algorithme de somme de contrôle pour vérifier l'intégrité de l'objet. MD5 n'est pas pris en charge pour les compartiments de répertoires. Vous pouvez spécifier l'un des algorithmes de hachage sécurisé (SHA) ou de contrôle de redondance cyclique (CRC) suivants pour vérifier l'intégrité des données :

- CRC32
- CRC32C
- SHA-1
- SHA-256

Vous pouvez utiliser l'API REST Amazon S3 ou les AWS SDK pour récupérer la valeur de la somme de contrôle pour des parties individuelles en utilisant `GetObject` ou. `HeadObject` Pour récupérer les valeurs de contrôle des parties individuelles de téléchargements partitionnés toujours en cours, vous pouvez utiliser `ListParts`.

> **Important**
>
> Lorsque vous utilisez les algorithmes de somme de contrôle précédents, les numéros de pièce en plusieurs parties doivent utiliser des numéros de pièce consécutifs. Si vous essayez

de traiter une demande de téléchargement partitionné avec des numéros de pièce non consécutifs, Amazon S3 génère une erreur `HTTP 400 Bad Request` (commande de pièces non valide).

Pour obtenir plus d'informations sur le fonctionnement des totaux de contrôle avec les objets en plusieurs parties, consultez Vérification de l'intégrité des objets.

Opérations simultanées de chargement partitionné

Dans un environnement de développement distribué, votre application peut lancer plusieurs mises à jour sur le même objet en même temps. Par exemple, votre application peut lancer plusieurs téléchargements partitionnés à l'aide de la même clé d'objet. Pour chacun de ces chargements, l'application peut ensuite charger des parties et envoyer une demande de chargement complet à Amazon S3 pour créer l'objet. Pour S3 Express One Zone, l'instant de création de l'objet correspond à la date d'achèvement du chargement partitionné.

Important

La gestion des versions n'est pas prise en charge pour les objets stockés dans des compartiments de répertoire.

Téléchargements en plusieurs parties et tarification

Lorsque vous lancez un chargement partitionné, Amazon S3 conserve toutes les parties jusqu'à ce que vous terminiez ou annuliez le chargement. Tout au long de sa durée de vie, le stockage, la bande passante et les demandes pour ce chargement partitionné ainsi que ses parties associées vous sont facturés. Si vous annulez le téléchargement en plusieurs parties, Amazon S3 supprime les artefacts de téléchargement et toutes les parties que vous avez chargées, et ils ne vous sont plus facturés. Aucuns frais de suppression anticipée ne sont facturés pour la suppression de téléchargements partitionnés incomplets, quelle que soit la classe de stockage spécifiée. Pour plus d'informations sur la tarification, consultez Tarification Amazon S3.

Important

Si la demande complète de téléchargement en plusieurs parties n'est pas envoyée avec succès, les parties de l'objet ne sont pas assemblées et aucun objet n'est créé. Vous êtes

facturé pour tout le stockage associé aux parties chargées. Il est important de terminer le téléchargement partitionné pour créer l'objet ou d'abandonner le téléchargement partitionné pour supprimer toutes les parties téléchargées.
Avant de pouvoir supprimer un compartiment de répertoire, vous devez terminer ou abandonner tous les téléchargements partitionnés en cours. Les compartiments d'annuaire ne prennent pas en charge les configurations S3 Lifecycle. Si nécessaire, vous pouvez répertorier vos téléchargements partitionnés actifs, puis annuler les téléchargements, puis supprimer votre bucket.

Opérations et autorisations de l'API de téléchargement en plusieurs parties

Pour autoriser l'accès aux opérations de l'API de gestion des objets sur un compartiment d'annuaire, vous accordez l'`s3express:CreateSession`autorisation dans une politique de compartiment ou une politique basée sur l'identité AWS Identity and Access Management (IAM).

Vous devez posséder les autorisations nécessaires pour utiliser les opérations de chargement partitionné. Vous pouvez utiliser des politiques de compartiment ou des politiques basées sur l'identité IAM pour accorder aux principaux IAM les autorisations nécessaires pour effectuer ces opérations. Le tableau suivant répertorie les autorisations nécessaires pour diverses opérations de chargement partitionné.

Vous pouvez identifier l'initiateur d'un téléchargement partitionné par le biais de l'`Initiator`élément. Si l'initiateur est un Compte AWS, cet élément fournit les mêmes informations que l'`0wner`élément. Si l'initiateur est un utilisateur IAM, cet élément fournit l'ARN utilisateur et le nom complet.

Action	Autorisations nécessaires
Créer un chargement partitionné	Pour créer le téléchargement partitionné, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket de répertoire.
Lancer un télécharg ement en plusieurs parties	Pour lancer le téléchargement partitionné, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket de répertoire.

Action	Autorisations nécessaires
Téléchargez une pièce	Pour télécharger une partie, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket du répertoire. Pour que l'initiateur puisse télécharger une partie, le propriétaire du compartiment doit autoriser l'initiateur à effectuer l'`s3express:CreateSession` action sur le compartiment de répertoire.
Télécharger une pièce (copie)	Pour télécharger une partie, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket du répertoire. Pour que l'initiateur puisse charger une partie pour un objet, le propriétaire du compartiment doit l'autoriser à effectuer l'action `s3express:CreateSession` sur l'objet.
Terminer un chargement partitionné	Pour effectuer un téléchargement partitionné, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket du répertoire. Pour que l'initiateur puisse effectuer un téléchargement en plusieurs parties, le propriétaire du compartiment doit autoriser l'initiateur à effectuer l'`s3express:CreateSession` action sur l'objet.
Abandonner un chargement partitionné	Pour annuler un téléchargement partitionné, vous devez être autorisé à effectuer l'`s3express:CreateSession` action. Pour que l'initiateur puisse abandonner un téléchargement partitionné, il doit disposer d'un accès autorisé explicite pour effectuer l'action. `s3express:CreateSession`
Lister les pièces	Pour répertorier les parties d'un téléchargement partitionné, vous devez être autorisé à effectuer l'`s3express:CreateSession` action sur le bucket du répertoire.
Lister des chargements partitionnés en cours	Pour répertorier les téléchargements partitionnés en cours vers un bucket, vous devez être autorisé à effectuer l'`s3:ListBucketMultipartUploads` action sur ce bucket.

Support des opérations d'API pour les téléchargements partitionnés

Les sections suivantes du manuel Amazon Simple Storage Service API Reference décrivent les opérations de l'API REST Amazon S3 pour les téléchargements partitionnés.

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- AbortMultipartUpload
- ListParts
- ListMultipartUploads

Exemples

Pour utiliser un téléchargement partitionné afin de télécharger un objet vers S3 Express One Zone dans un compartiment de répertoire, consultez les exemples suivants.

Rubriques

- Création d’un chargement partitionné
- Chargement des parties d'un téléchargement en plusieurs parties
- Achèvement d’un chargement partitionné
- Interruption d'un chargement partitionné
- Création d’une opération de copie d’un chargement partitionné
- Liste des téléchargements partitionnés en cours
- Répertorier les parties d'un téléchargement en plusieurs parties

Création d’un chargement partitionné

Les exemples suivants montrent comment créer un téléchargement partitionné.

Utilisation des AWS SDK

SDK for Java 2.x

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
 * all the upload parts
 *
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--use1-az4--x-s3'
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {

     CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
             .bucket(bucketName)
             .key(key)
             .build();

     String uploadId = null;

     try {
         CreateMultipartUploadResponse response =
 s3.createMultipartUpload(createMultipartUploadRequest);
         uploadId = response.uploadId();
     }
     catch (S3Exception e) {
         System.err.println(e.awsErrorDetails().errorMessage());
         System.exit(1);
     }
     return uploadId;
```

SDK for Python

Example

```
def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
```

```
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :return: The UploadId for the multipart upload if created successfully, else None
    '''

    try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
key_name)
        return mpu['UploadId']
     except ClientError as e:
        logging.error(e)
        return None
```

À l'aide du AWS CLI

Cet exemple montre comment créer un téléchargement partitionné vers un bucket de répertoire à l'aide du AWS CLI. *Cette commande lance un téléchargement en plusieurs parties vers le bucket de répertoire bucket-base-name-- azid --x-s3 pour l'objet KEY_NAME.* Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api create-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Pour plus d'informations, consultez create-multipart-uploadle AWS Command Line Interface.

Chargement des parties d'un téléchargement en plusieurs parties

Les exemples suivants montrent comment télécharger des parties d'un téléchargement partitionné.

Utilisation des AWS SDK

SDK for Java 2.x

L'exemple suivant montre comment diviser un seul objet en plusieurs parties, puis télécharger ces parties dans un bucket de répertoire à l'aide du SDK for Java 2.x.

Example

```
/**
```

```
* This method creates part requests and uploads individual parts to S3 and then
returns all the completed parts
*
* @param s3
* @param bucketName
* @param key
* @param uploadId
* @throws IOException
*/
private static ListCompletedPartmultipartUpload(S3Client s3, String bucketName,
String key, String uploadId, String filePath) throws IOException {

        int partNumber = 1;
        ListCompletedPart completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        // read the local file, breakdown into chunks and process
        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            int position = 0;
            while (position < fileSize) {
                file.seek(position);
                int read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .partNumber(partNumber)
                        .build();

                UploadPartResponse partResponse = s3.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
```

```
                position += read;
                partNumber++;
            }
        }

        catch (IOException e) {
            throw e;
        }
        return completedParts;
    }
```

SDK for Python

L'exemple suivant montre comment diviser un seul objet en plusieurs parties, puis télécharger ces parties dans un bucket de répertoire à l'aide du SDK pour Python.

Example

```
def multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_size):
    '''
    Break up a file into multiple parts and upload those parts to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name for object to be uploaded and for the local file
 that's being uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart upload if all parts are uploaded
 successfully, else None
    '''

    part_list = []
    try:
        with open(key_name, 'rb') as file:
            part_counter = 1
            while True:
                file_part = file.read(part_size)
                if not len(file_part):
                    break
                upload_part = s3_client.upload_part(
                    Bucket = bucket_name,
```

```
                    Key = key_name,
                    UploadId = mpu_id,
                    Body = file_part,
                    PartNumber = part_counter
                )
                part_list.append({'PartNumber': part_counter, 'ETag':
upload_part['ETag']})
                part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list
```

À l'aide du AWS CLI

Cet exemple montre comment diviser un seul objet en plusieurs parties, puis télécharger ces parties dans un bucket de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api upload-part --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --part-number 1 --body LOCAL_FILE_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Pour plus d'informations, consultez la section upload-part dans le. AWS Command Line Interface

Achèvement d’un chargement partitionné

Les exemples suivants montrent comment effectuer un téléchargement partitionné.

Utilisation des AWS SDK

SDK for Java 2.x

Les exemples suivants montrent comment effectuer un téléchargement partitionné à l'aide du SDK for Java 2.x.

Example

```
/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3
```

```
* @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
* @param key
* @param uploadId
* @param uploadParts
*/
private static void completeMultipartUpload(S3Client s3, String bucketName, String
key, String uploadId, ListCompletedPart uploadParts) {
       CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
               .parts(uploadParts)
               .build();

       CompleteMultipartUploadRequest completeMultipartUploadRequest =
               CompleteMultipartUploadRequest.builder()
                       .bucket(bucketName)
                       .key(key)
                       .uploadId(uploadId)
                       .multipartUpload(completedMultipartUpload)
                       .build();

       s3.completeMultipartUpload(completeMultipartUploadRequest);
   }

   public static void multipartUploadTest(S3Client s3, String bucketName, String
key, String localFilePath)  {
       System.out.println("Starting multipart upload for: " + key);
       try {
           String uploadId = createMultipartUpload(s3, bucketName, key);
           System.out.println(uploadId);
           ListCompletedPart parts = multipartUpload(s3, bucketName, key, uploadId,
localFilePath);
           completeMultipartUpload(s3, bucketName, key, uploadId, parts);
           System.out.println("Multipart upload completed for: " + key);
       }

       catch (Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

SDK for Python

Les exemples suivants montrent comment effectuer un téléchargement en plusieurs parties à l'aide du SDK pour Python.

Example

```
def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: The list of uploaded part numbers with their associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'OBJECT_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_upload(s3_client, bucket_name, key_name, mpu_id,
 part_size)
```

```
        if part_list is not None:
            if complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id,
part_list):
                print (f'{key_name} successfully uploaded through a ultipart upload
to {bucket_name}')
            else:
                print (f'Could not upload {key_name} hrough a multipart upload to
{bucket_name}')
```

À l'aide du AWS CLI

Cet exemple montre comment effectuer un téléchargement partitionné pour un bucket de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
 --multipart-upload file://parts.json
```

Cet exemple utilise une structure JSON qui décrit les parties du téléchargement partitionné qui doivent être réassemblées dans le fichier complet. Dans cet exemple, le `file://` préfixe est utilisé pour charger la structure JSON à partir d'un fichier du dossier local nommé`parts`.

parts.json :

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Pour plus d'informations, consultez complete-multipart-uploadle AWS Command Line Interface.

Interruption d'un chargement partitionné

Les exemples suivants montrent comment annuler un téléchargement partitionné.

Utilisation des AWS SDK

SDK for Java 2.x

L'exemple suivant montre comment annuler un téléchargement partitionné à l'aide du SDK pour Java 2.x.

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {

        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
 ListMultipartUploadsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListMultipartUploadsResponse response =
 s3.listMultipartUploads(listMultipartUploadsRequest);
            ListMultipartUpload uploads = response.uploads();

            AbortMultipartUploadRequest abortMultipartUploadRequest;
            for (MultipartUpload upload: uploads) {
                abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(upload.key())
                        .uploadId(upload.uploadId())
                        .build();

                s3.abortMultipartUpload(abortMultipartUploadRequest);
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

L'exemple suivant montre comment annuler un téléchargement partitionné à l'aide du SDK pour Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError


def abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
    '''
    Aborts a partial multipart upload in a directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket where the multipart upload was initiated - for
 example, 'doc-example-bucket--usw2-az1--x-s3'
    :param key_name: Name of the object for which the multipart upload needs to be
 aborted
    :param upload_id: Multipart upload ID for the multipart upload to be aborted
    :return: True if the multipart upload was successfully aborted, False if not
    '''
    try:
        s3_client.abort_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = upload_id
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
        upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    if abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
        print (f'Multipart upload for object {key_name} in {bucket_name} bucket has
 been aborted')
    else:
        print (f'Unable to abort multipart upload for object {key_name} in
 {bucket_name} bucket')
```

À l'aide du AWS CLI

L'exemple suivant montre comment annuler un téléchargement partitionné à l'aide du. AWS CLI Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
 --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB0OxUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Pour plus d'informations, consultez abort-multipart-uploadle AWS Command Line Interface.

Création d'une opération de copie d'un chargement partitionné

Les exemples suivants montrent comment copier des objets d'un compartiment à un autre à l'aide d'un téléchargement partitionné.

Utilisation des AWS SDK

SDK for Java 2.x

L'exemple suivant montre comment utiliser un téléchargement partitionné pour copier par programmation un objet d'un bucket à un autre à l'aide du SDK pour Java 2.x.

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
 * all the upload parts.
 *
 * @param s3
 * @param bucketName
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {
        CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
```

```
                .build();
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
 }

 /**
  * Creates copy parts based on source object size and copies over individual parts
  *
  * @param s3
  * @param sourceBucket
  * @param sourceKey
  * @param destnBucket
  * @param destnKey
  * @param uploadId
  * @return
  * @throws IOException
  */
   public static ListCompletedPart multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

        // Get the object size to track the end of the copy operation.
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(sourceBucket)
                .key(sourceKey)
                .build();
        HeadObjectResponse response = s3.headObject(headObjectRequest);
        Long objectSize = response.contentLength();

        System.out.println("Source Object size: " + objectSize);

        // Copy the object using 20 MB parts.
        long partSize = 20 * 1024 * 1024;
        long bytePosition = 0;
        int partNum = 1;
```

```
        ListCompletedPart completedParts = new ArrayList<>();
        while (bytePosition < objectSize) {
            // The last part might be smaller than partSize, so check to make sure
            // that lastByte isn't beyond the end of the object.
            long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

            System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

            // Copy this part.
            UploadPartCopyRequest req = UploadPartCopyRequest.builder()
                    .uploadId(uploadId)
                    .sourceBucket(sourceBucket)
                    .sourceKey(sourceKey)
                    .destinationBucket(destnBucket)
                    .destinationKey(destnKey)
                    .copySourceRange("bytes="+bytePosition+"-"+lastByte)
                    .partNumber(partNum)
                    .build();
            UploadPartCopyResponse res = s3.uploadPartCopy(req);
            CompletedPart part = CompletedPart.builder()
                    .partNumber(partNum)
                    .eTag(res.copyPartResult().eTag())
                    .build();
            completedParts.add(part);
            partNum++;
            bytePosition += partSize;
        }
        return completedParts;
    }


    public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
srcKey, String destnBucket, String destnKey)  {
        System.out.println("Starting multipart copy for: " + srcKey);
        try {
            String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUploadCopy(s3, srcBucket,
srcKey,destnBucket,  destnKey, uploadId);
            completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
            System.out.println("Multipart copy completed for: " + srcKey);
        } catch (Exception e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }
```

SDK for Python

L'exemple suivant montre comment utiliser un téléchargement partitionné pour copier par programmation un objet d'un bucket à un autre à l'aide du SDK pour Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def head_object(s3_client, bucket_name, key_name):
    '''
    Returns metadata for an object in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains the object to query for metadata
    :param key_name: Key name to query for metadata
    :return: Metadata for the specified object if successful, else None
    '''

    try:
        response = s3_client.head_object(
            Bucket = bucket_name,
            Key = key_name
        )
        return response
    except ClientError as e:
        logging.error(e)
        return None

def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :return: UploadId for the multipart upload if created successfully, else None
    '''
```

```
    try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
 key_name)
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None

def multipart_copy_upload(s3_client, source_bucket_name, key_name,
 target_bucket_name, mpu_id, part_size):
    '''
    Copy an object in a directory bucket to another bucket in multiple parts of a
 specified size

    :param s3_client: boto3 S3 client
    :param source_bucket_name: Bucket where the source object exists
    :param key_name: Key name of the object to be copied
    :param target_bucket_name: Destination bucket for copied object
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart copy if all parts are copied successfully,
 else None
    '''

    part_list = []
    copy_source = {
        'Bucket': source_bucket_name,
        'Key': key_name
    }
    try:
        part_counter = 1
        object_size = head_object(s3_client, source_bucket_name, key_name)
        if object_size is not None:
            object_size = object_size['ContentLength']
        while (part_counter - 1) * part_size <object_size:
            bytes_start = (part_counter - 1) * part_size
            bytes_end = (part_counter * part_size) - 1
            upload_copy_part = s3_client.upload_part_copy (
                Bucket = target_bucket_name,
                CopySource = copy_source,
                CopySourceRange = f'bytes={bytes_start}-{bytes_end}',
```

```
                Key = key_name,
                PartNumber = part_counter,
                UploadId = mpu_id
            )
            part_list.append({'PartNumber': part_counter, 'ETag':
 upload_copy_part['CopyPartResult']['ETag']})
            part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list

def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: List of uploaded part numbers with associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    source_bucket_name = 'SOURCE_BUCKET_NAME'
    target_bucket_name = 'TARGET_BUCKET_NAME'
    key_name = 'KEY_NAME'
```

```
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, target_bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_copy_upload(s3_client, source_bucket_name, key_name,
target_bucket_name, mpu_id, part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, target_bucket_name, key_name,
mpu_id, part_list):
                print (f'{key_name} successfully copied through multipart copy from
{source_bucket_name} to {target_bucket_name}')
            else:
                print (f'Could not copy {key_name} through multipart copy from
{source_bucket_name} to {target_bucket_name}')
```

À l'aide du AWS CLI

L'exemple suivant montre comment utiliser un téléchargement partitionné pour copier par programmation un objet d'un compartiment vers un compartiment de répertoire à l'aide du. AWS CLI Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api upload-part-copy --bucket bucket-base-name--azid--x-s3 --key TARGET_KEY_NAME
 --copy-source SOURCE_BUCKET_NAME/SOURCE_KEY_NAME --part-number 1 --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Pour plus d'informations, consultez upload-part-copyle AWS Command Line Interface.

Liste des téléchargements partitionnés en cours

Pour répertorier les téléchargements partitionnés en cours vers un bucket de répertoire, vous pouvez utiliser les AWS SDK ou le. AWS CLI

Utilisation des AWS SDK

SDK for Java 2.x

Les exemples suivants montrent comment répertorier les téléchargements partitionnés en cours (incomplets) à l'aide du SDK pour Java 2.x.

Example

```
public static void listMultiPartUploads( S3Client s3, String bucketName) {
       try {
           ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
               .bucket(bucketName)
               .build();

           ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
           List MultipartUpload uploads = response.uploads();
           for (MultipartUpload upload: uploads) {
               System.out.println("Upload in progress: Key = \"" + upload.key() +
"\", id = " + upload.uploadId());
           }
     }
     catch (S3Exception e) {
           System.err.println(e.getMessage());
           System.exit(1);
     }
 }
```

SDK for Python

Les exemples suivants montrent comment répertorier les téléchargements partitionnés en cours (incomplets) à l'aide du SDK pour Python.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_multipart_uploads(s3_client, bucket_name):
    '''
    List any incomplete multipart uploads in a directory bucket in e specified gion

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to check for incomplete multipart uploads
    :return: List of incomplete multipart uploads if there are any, None if not
    '''
```

```
    try:
        response = s3_client.list_multipart_uploads(Bucket = bucket_name)
        if 'Uploads' in response.keys():
            return response['Uploads']
        else:
            return None
    except ClientError as e:
        logging.error(e)

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
    multipart_uploads = list_multipart_uploads(s3_client, bucket_name)
    if multipart_uploads is not None:
        print (f'There are {len(multipart_uploads)} ncomplete multipart uploads for
 {bucket_name}')
    else:
        print (f'There are no incomplete multipart uploads for {bucket_name}')
```

À l'aide du AWS CLI

Les exemples suivants montrent comment répertorier les téléchargements partitionnés en cours (incomplets) à l'aide du. AWS CLI Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--azid--x-s3
```

Pour plus d'informations, consultez list-multipart-uploadsle AWS Command Line Interface.

Répertorier les parties d'un téléchargement en plusieurs parties

Les exemples suivants montrent comment répertorier les parties d'un téléchargement partitionné vers un bucket de répertoire.

Utilisation des AWS SDK

SDK for Java 2.x

> Les exemples suivants montrent comment répertorier les parties d'un téléchargement partitionné vers un bucket de répertoire à l'aide du SDK for Java 2.x.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
 objKey, String uploadID) {

        try {
            ListPartsRequest listPartsRequest = ListPartsRequest.builder()
                .bucket(bucketName)
                .uploadId(uploadID)
                .key(objKey)
                .build();

            ListPartsResponse response = s3.listParts(listPartsRequest);
            ListPart parts = response.parts();
            for (Part part: parts) {
                System.out.println("Upload in progress: Part number = \"" +
 part.partNumber() + "\", etag = " + part.eTag());
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }


    }
```

SDK for Python

Les exemples suivants montrent comment répertorier les parties d'un téléchargement partitionné vers un bucket de répertoire à l'aide du SDK pour Python.

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_parts(s3_client, bucket_name, key_name, upload_id):
    '''
    Lists the parts that have been uploaded for a specific multipart upload to a
 directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that multipart uploads parts have been uploaded to
```

```
    :param key_name: Name of the object that has parts uploaded
    :param upload_id: Multipart upload ID that the parts are associated with
    :return: List of parts associated with the specified multipart upload, None if
 there are no parts
    '''
    parts_list = []
    next_part_marker = ''
    continuation_flag = True
    try:
        while continuation_flag:
            if next_part_marker == '':
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id
                )
            else:
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id,
                    NextPartMarker = next_part_marker
                )
            if 'Parts' in response:
                for part in response['Parts']:
                    parts_list.append(part)
                if response['IsTruncated']:
                    next_part_marker = response['NextPartNumberMarker']
                else:
                    continuation_flag = False
            else:
                continuation_flag = False
        return parts_list
    except ClientError as e:
        logging.error(e)
        return None

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    parts_list = list_parts(s3_client, bucket_name, key_name, upload_id)
```

```
    if parts_list is not None:
        print (f'{key_name} has {len(parts_list)} parts uploaded to {bucket_name}')
    else:
        print (f'There are no multipart uploads with that upload ID for
{bucket_name} bucket')
```

À l'aide du AWS CLI

Les exemples suivants montrent comment répertorier les parties d'un téléchargement partitionné vers un bucket de répertoire à l'aide du AWS CLI. Pour utiliser la commande, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
aws s3api list-parts --bucket bucket-base-name--azid--x-s3 --key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Pour plus d'informations, voir list-parts dans le. AWS Command Line Interface

Copie d’un objet vers un compartiment de répertoires

L'opération de copie crée une copie d'un objet déjà stocké dans Amazon S3. Vous pouvez copier des objets entre des compartiments de répertoires et des compartiments à usage général. Vous pouvez également copier des objets au sein d’un compartiment et entre des compartiments du même type, par exemple, d’un compartiment de répertoires à un autre.

Vous pouvez créer une copie d'un objet d'une capacité maximale de 5 Go en une seule opération atomique. Toutefois, pour copier un objet dont la taille est supérieure à 5 Go, vous devez utiliser les opérations de l'API de téléchargement partitionné. Pour plus d’informations, consultez Utilisation de téléchargements partitionnés avec des compartiments de répertoires.

Autorisations

Pour copier des objets, vous devez disposer des autorisations suivantes :

- Pour copier des objets d’un compartiment de répertoires vers un autre, vous devez disposer de l’autorisation `s3express:CreateSession`.
- Pour copier des objets à partir de compartiments de répertoires vers des compartiments à usage général, vous devez disposer de l’autorisation `s3express:CreateSession` et de l’autorisation `s3:PutObject` permettant d’écrire la copie des objets dans le compartiment de destination.

- Pour copier des objets depuis des compartiments à usage général vers des compartiments de répertoire, vous devez disposer de l'`s3express:CreateSession`autorisation et de `s3:GetObject` l'autorisation nécessaires pour lire l'objet source copié.

  Pour plus d'informations, veuillez consulter CopyObject dans la Référence d'API Amazon Simple Storage Service.

Chiffrement

Amazon S3 chiffre automatiquement tous les nouveaux objets chargés dans un compartiment S3. La configuration de chiffrement par défaut d'un compartiment S3 est toujours activée et est au minimum définie sur le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3).

Pour les compartiments de répertoire, seul le SSE-S3 est pris en charge. Pour les compartiments à usage général, vous pouvez utiliser le SSE-S3 (par défaut), le chiffrement côté serveur avec des clés () AWS Key Management Service (SSE-KMS AWS KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C).

Si vous faites une demande de copie qui définit les paramètres SSE-C, SSE-KMS ou DSSE-KMS d'un bucket d'annuaire comme source ou destination, la réponse renvoie une erreur,

Balises

Les compartiments de répertoires ne prennent pas en charge les balises. Si vous copiez un objet contenant des balises d'un compartiment à usage général vers un compartiment de répertoire, vous recevez une `501 (Not Implemented)` réponse HTTP. Pour plus d'informations, veuillez consulter CopyObject dans la Référence d'API Amazon Simple Storage Service.

ETags

Les étiquettes d'entité (ETags) pour S3 Express One Zone sont des chaînes alphanumériques aléatoires et ne sont pas des checksums MD5. Pour garantir l'intégrité des objets, utilisez des checksums supplémentaires.

Sommes de contrôle supplémentaires

S3 Express One Zone vous offre la possibilité de choisir l'algorithme de somme de contrôle utilisé pour valider vos données pendant le chargement ou le téléchargement. Vous pouvez sélectionner l'un des algorithmes de contrôle d'intégrité des données Secure Hash Algorithms (SHA) ou Cyclic

Redundancy Check (CRC) suivants : CRC32, CRC32C, SHA-1 ou SHA-256. Les checksums basés sur MD5 ne sont pas pris en charge avec la classe de stockage S3 Express One Zone.

Pour plus d'informations, consultez Bonnes pratiques supplémentaires en matière de somme de contrôle S3.

Fonctionnalités prises en charge

Pour plus d'informations sur les fonctionnalités Amazon S3 prises en charge pour S3 Express One Zone, consultezEn quoi S3 Express One Zone est-il différent ?.

Utilisation de la console S3 (copie vers un compartiment de répertoires)

Pour copier un objet d'un bucket à usage général ou d'un bucket de répertoire vers un bucket de répertoire

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez le compartiment à partir duquel vous souhaitez copier des objets :
   - Pour effectuer une copie à partir d'un bucket à usage général, choisissez l'onglet Buckets à usage général.
   - Pour effectuer une copie à partir d'un bucket de répertoire, choisissez l'onglet Buckets de répertoire.
4. Choisissez le compartiment à usage général ou le compartiment de répertoire contenant les objets que vous souhaitez copier.
5. Cliquez sur l'onglet Objets. Sur la page Objets, cochez la case située à gauche des noms des objets que vous souhaitez copier.
6. Dans le menu Actions, choisissez Copy (Copier).

   La page Copier apparaît.
7. Sous Destination, choisissez Directory bucket pour votre type de destination. Pour spécifier le chemin de destination, choisissez Parcourir S3, naviguez jusqu'à la destination, puis cliquez sur le bouton d'option situé à gauche de la destination. Choisissez Choose destination (Choisir une destination) en bas à droite.

   Vous pouvez également saisir le chemin de destination.

8. Sous Sommes de contrôle, indiquez si vous souhaitez copier les objets avec leurs fonctions de somme de contrôle existantes ou remplacer les fonctions de somme de contrôle existantes par une nouvelle. Lorsque vous avez chargé les objets, vous aviez la possibilité de spécifier l'algorithme total de contrôle utilisé pour vérifier l'intégrité des données. Lors de la copie de l'objet, vous avez la possibilité de sélectionner une nouvelle fonction. Si vous n'avez pas initialement spécifié de somme de contrôle supplémentaire, vous pouvez utiliser la section Sommes de contrôle pour en ajouter une.

   > **Note**
   >
   > Même si vous choisissez d'utiliser la même fonction de somme de contrôle, la valeur de votre somme de contrôle peut changer si la taille de l'objet est supérieure à 16 Mo. La valeur du total de contrôle peut changer en raison de la façon dont les totaux de contrôle sont calculés pour les chargements partitionnés. Pour plus d'informations sur la façon dont le total de contrôle peut changer lors de la copie de l'objet, consultez Utilisation de totaux de contrôle au niveau des parties pour les chargements partitionnés.

   Pour modifier la fonction de total de contrôle, sélectionnez Replace with a new checksum function (Remplacer par une nouvelle fonction de total de contrôle). Choisissez la nouvelle fonction de somme de contrôle dans la liste déroulante. Lorsque l'objet est copié, la nouvelle somme de contrôle est calculée et stockée à l'aide de l'algorithme spécifié.
9. En bas à droite, choisissez Copy (Copier). Amazon S3 copie votre objet dans la destination.

Utilisation de la console S3 (copie vers un compartiment à usage général)

Pour copier un objet d'un compartiment de répertoires vers un compartiment à usage général

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l'onglet Buckets du répertoire.
4. Choisissez le compartiment de répertoire qui contient les objets que vous souhaitez copier.
5. Cliquez sur l'onglet Objets. Sur la page Objets, cochez la case située à gauche des noms des objets que vous souhaitez copier.
6. Dans le menu Actions, choisissez Copy (Copier).

7. Sous Destination, choisissez Bucket à usage général pour votre type de destination. Pour spécifier le chemin de destination, choisissez Browse S3, naviguez jusqu'à la destination, puis cliquez sur le bouton d'option situé à gauche de la destination. Choisissez Choose destination (Choisir une destination) en bas à droite.

   Vous pouvez également saisir le chemin de destination.

8. Sous Sommes de contrôle, indiquez si vous souhaitez copier les objets avec leurs fonctions de somme de contrôle existantes ou remplacer les fonctions de somme de contrôle existantes par une nouvelle. Lorsque vous avez chargé les objets, vous aviez la possibilité de spécifier l'algorithme total de contrôle utilisé pour vérifier l'intégrité des données. Lors de la copie de l'objet, vous avez la possibilité de sélectionner une nouvelle fonction. Si vous n'avez pas initialement spécifié de somme de contrôle supplémentaire, vous pouvez utiliser la section Sommes de contrôle pour en ajouter une.

   > **Note**
   >
   > Même si vous choisissez d'utiliser la même fonction de somme de contrôle, la valeur de votre somme de contrôle peut changer si la taille de l'objet est supérieure à 16 Mo. La valeur du total de contrôle peut changer en raison de la façon dont les totaux de contrôle sont calculés pour les chargements partitionnés. Pour plus d'informations sur la façon dont le total de contrôle peut changer lors de la copie de l'objet, consultez Utilisation de totaux de contrôle au niveau des parties pour les chargements partitionnés.

   Pour modifier la fonction de total de contrôle, sélectionnez Replace with a new checksum function (Remplacer par une nouvelle fonction de total de contrôle). Choisissez la nouvelle fonction de somme de contrôle dans la liste déroulante. Lorsque l'objet est copié, la nouvelle somme de contrôle est calculée et stockée à l'aide de l'algorithme spécifié.

9. En bas à droite, choisissez Copy (Copier). Amazon S3 copie votre objet dans la destination.

Utilisation des AWS SDK

SDK for Java 2.x

Example

```
public static void copyBucketObject (S3Client s3, String sourceBucket, String
objectKey, String targetBucket) {
     CopyObjectRequest copyReq = CopyObjectRequest.builder()
         .sourceBucket(sourceBucket)
         .sourceKey(objectKey)
         .destinationBucket(targetBucket)
         .destinationKey(objectKey)
         .build();
      String temp = "";

      try {
          CopyObjectResponse copyRes = s3.copyObject(copyReq);
          System.out.println("Successfully copied " + objectKey +" from bucket " +
sourceBucket +" into bucket "+targetBucket);
      }

      catch (S3Exception e) {
          System.err.println(e.awsErrorDetails().errorMessage());
          System.exit(1);
      }
}
```

En utilisant le AWS CLI

L'`copy-object`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour copier un objet d'un compartiment vers un autre. Vous pouvez copier des objets entre les types de compartiments. Pour exécuter cette commande, remplacez les espaces réservés saisis par l'utilisateur par vos propres informations.

```
aws s3api copy-object --copy-source bucket SOURCE_BUCKET/SOURCE_KEY_NAME --
key TARGET_KEY_NAME --bucket TARGET_BUCKET_NAME
```

Pour plus d’informations, consultez la section copy-object dans la référence des commandes AWS CLI .

Suppression d’un objet dans un compartiment de répertoires

Vous pouvez supprimer des objets d'un compartiment d'annuaire Amazon S3 à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) ou AWS des kits SDK. Pour plus d’informations, consultez Compartiments de répertoire et Qu’est-ce que S3 Express One Zone ?.

Warning

- La suppression d'un objet ne peut pas être annulée.
- Cette action supprime tous les objets spécifiés. Lorsque vous supprimez des dossiers, attendez la fin de l'action de suppression pour ajouter de nouveaux objets au dossier. Dans le cas contraire, de nouveaux objets pourraient également être supprimés.

Note

Lorsque vous supprimez par programmation plusieurs objets d'un bucket de répertoire, notez ce qui suit :

- Les clés d’objet figurant dans les demandes `DeleteObjects` doivent contenir au moins un caractère autre qu’une espace. Les chaînes contenant tous les espaces blancs ne sont pas prises en charge.
- Les clés d'objet des `DeleteObjects` demandes ne peuvent pas contenir de caractères de contrôle Unicode, à l'exception de newline (`\n`), tab (`\t`) et carrierreturn (`\r`).

Utilisation de la console S3

Pour supprimer des objets

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez l’onglet Compartiments de répertoires.
4. Choisissez le compartiment de répertoire qui contient les objets que vous souhaitez supprimer.
5. Cliquez sur l'onglet Objets. Dans la liste des objets, cochez la case située à gauche de l'objet ou des objets que vous souhaitez supprimer.

6. Sélectionnez Delete (Supprimer).

7. Sur la page Supprimer des objets, entrez **`permanently delete`** dans la zone de texte.

8. Choisissez Supprimer les objets.

Utilisation des AWS SDK

SDK for Java 2.x

Example

L'exemple suivant supprime des objets d'un bucket de répertoire à l'aide du AWS SDK for Java 2.x.

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {



        try {

            DeleteObjectRequest del = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();

            s3Client.deleteObject(del);

            System.out.println("Object " + objectKey + " has been deleted");


        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

    }
```

SDK for Python

Example

L'exemple suivant supprime des objets d'un bucket de répertoire à l'aide du AWS SDK for Python (Boto3).

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_objects(s3_client, bucket_name, objects):
    '''
    Delete a list of objects in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains objects to be deleted; for example,
 'doc-example-bucket--usw2-az1--x-s3'
    :param objects: List of dictionaries that specify the key names to delete
    :return: Response output, else False
    '''

    try:
        response = s3_client.delete_objects(
            Bucket = bucket_name,
            Delete = {
                'Objects': objects
            }
        )
        return response
    except ClientError as e:
        logging.error(e)
        return False


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    objects = [
        {
            'Key': '0.txt'
        },
        {
```

```
            'Key': '1.txt'
        },
        {
            'Key': '2.txt'
        },
        {
            'Key': '3.txt'
        },
        {
            'Key': '4.txt'
        }
    ]

    s3_client = boto3.client('s3', region_name = region)
    results = delete_objects(s3_client, bucket_name, objects)
    if results is not None:
        if 'Deleted' in results:
            print (f'Deleted {len(results["Deleted"])} objects from {bucket_name}')
        if 'Errors' in results:
            print (f'Failed to delete {len(results["Errors"])} objects from
{bucket_name}')
```

À l'aide du AWS CLI

L'`delete-object`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour supprimer un objet d'un bucket de répertoire. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api delete-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Pour plus d'informations, consultez la section delete-object dans la référence des commandes AWS CLI .

Téléchargement d'un objet dans un bucket de répertoire

Les exemples de code suivants montrent comment lire des données depuis (télécharger) un objet dans un bucket d'annuaire Amazon S3 à l'aide de l'opération `GetObject` API.

Utilisation des AWS SDK

SDK for Java 2.x

Example

L'exemple de code suivant montre comment lire les données d'un objet dans un bucket de répertoire à l'aide du AWS SDK for Java 2.x.

```
public static void getObject(S3Client s3Client, String bucketName, String objectKey)
 {
     try {
         GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(objectKey)
            .bucket(bucketName)
            .build();

         ResponseBytes GetObjectResponse objectBytes =
 s3Client.getObjectAsBytes(objectRequest);
         byte[] data = objectBytes.asByteArray();

         //Print object contents to console
         String s = new String(data, StandardCharsets.UTF_8);
         System.out.println(s);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

L'exemple de code suivant montre comment lire les données d'un objet dans un bucket de répertoire à l'aide du AWS SDK for Python (Boto3).

```
import boto3
from botocore.exceptions import ClientError
from botocore.response import StreamingBody
```

```
def get_object(s3_client: boto3.client, bucket_name: str, key_name: str) ->
 StreamingBody:
    """
    Gets the object.
    :param s3_client:
    :param bucket_name: The bucket that contains the object.
    :param key_name: The key of the object to be downloaded.
    :return: The object data in bytes.
    """
    try:
        response = s3_client.get_object(Bucket=bucket_name, Key=key_name)
        body = response['Body'].read()
        print(f"Got object '{key_name}' from bucket '{bucket_name}'.")
    except ClientError:
        print(f"Couldn't get object '{key_name}' from bucket '{bucket_name}'.")
        raise
    else:
        return body

def main():
    s3_client = boto3.client('s3')
    resp = get_object(s3_client, 'doc-example-bucket--use1-az4--x-s3', 'sample.txt')
    print(resp)

if __name__ == "__main__":
     main()
```

À l'aide du AWS CLI

L'exemple de commande `get-object` suivant montre comment utiliser l'interface AWS CLI pour télécharger un objet depuis Amazon S3. Cette commande extrait l'objet *KEY_NAME* du bucket du répertoire*bucket-base-name*`--`*azid*`--x-s3`. L'objet sera téléchargé dans un fichier nommé *LOCAL_FILE_NAME*. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3api get-object --bucket bucket-base-name--azid--x-s3 --
key KEY_NAME LOCAL_FILE_NAME
```

Pour plus d'informations, consultez la section get-object dans la référence des commandes AWS CLI .

Utilisation **`HeadObject`** avec des buckets de répertoires

Les exemples de AWS SDK et de AWS CLI suivants montrent comment utiliser l'opération `HeadObject` API pour récupérer les métadonnées d'un objet dans un bucket d'annuaire Amazon S3 sans renvoyer l'objet lui-même.

Utilisation des AWS SDK

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String
 objectKey) {
    try {
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with
 ETag: \"%s\"", objectKey, bucketName, response.eTag());
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
```

À l'aide du AWS CLI

L'`head-object`exemple de commande suivant montre comment vous pouvez utiliser le AWS CLI pour récupérer les métadonnées d'un objet. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api head-object --bucket bucket-base-name--azid--x-s3 --key KEY_NAME
```

Pour plus d’informations, consultez la section head-object dans la référence des commandes AWS CLI .

Sécurité pour S3 Express One Zone

Chez AWS, la sécurité dans le cloud est la priorité numéro 1. En tant que client AWS, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité. La sécurité est une responsabilité partagée entre AWS et vous-même. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud – AWS est responsable de la protection de l'infrastructure qui exécute des Services AWS dans le AWS Cloud. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des AWS Compliance Programs.

  Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon S3 Express One Zone, consultez Services AWS in Scope by Compliance Program.
- Sécurité dans le cloud – Votre responsabilité est fonction du Service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de S3 Express One Zone. Les rubriques suivantes montrent comment configurer S3 Express One Zone pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres Services AWS qui vous aideront à surveiller et sécuriser vos ressources dans le cadre de l'utilisation de S3 Express One Zone.

Rubriques

Protection et chiffrement des données

Pour plus d'informations sur la façon dont S3 Express One Zone chiffre et protège les données, consultez les rubriques suivantes.

Rubriques

Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)

Par défaut, tous les objets stockés dans les compartiments de répertoires sont automatiquement chiffrés à l'aide d'un chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3). Les chargements non chiffrés vers des compartiments de répertoires ne sont pas autorisés. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) et Protection des données à l'aide du chiffrement.

Les compartiments de répertoires ne prennent pas en charge le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement double couche côté serveur avec des clés AWS Key Management Service (AWS KMS) (DSSE-KMS) ni le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C).

Chiffrement en transit

S3 Express One Zone est accessible uniquement via HTTPS (TLS).

S3 Express One Zone utilise des points de terminaison d'API régionaux et zonaux. Selon l'opération d'API Amazon S3 que vous utilisez, un point de terminaison régional ou zonal est requis. Vous pouvez accéder aux points de terminaison zonaux et régionaux via un point de terminaison de cloud privé virtuel (VPC) de passerelle. Il n'y a pas de frais supplémentaires pour l'utilisation de points de terminaison de passerelle. Pour en savoir plus sur les points de terminaison d'API régionaux et zonaux, consultez Mise en réseau pour S3 Express One Zone.

Sommes de contrôle supplémentaires

S3 Express One Zone vous offre la possibilité de choisir l'algorithme de somme de contrôle utilisé pour valider vos données pendant le chargement ou le téléchargement. Vous pouvez sélectionner

l'un des algorithmes de contrôle d'intégrité des données Secure Hash Algorithms (SHA) ou Cyclic Redundancy Check (CRC) suivants : CRC32, CRC32C, SHA-1 ou SHA-256. Les checksums basés sur MD5 ne sont pas pris en charge avec la classe de stockage S3 Express One Zone.

Pour plus d'informations, consultez Bonnes pratiques supplémentaires en matière de somme de contrôle S3.

Suppression de données

Vous pouvez supprimer un ou plusieurs objets directement dans S3 Express One Zone à l'aide de la console Amazon S3, des kits AWS SDK, de l'AWS Command Line Interface (AWS CLI) ou de l'API REST Amazon S3. Étant donné que tous les objets figurant dans les compartiments de répertoires entraînent des coûts de stockage, nous vous recommandons de supprimer les objets dont vous n'avez plus besoin.

La suppression d'un objet stocké dans un compartiment de répertoires supprime également de manière récursive tous les répertoires parents, si ces répertoires parents ne contiennent aucun objet autre que l'objet en cours de suppression.

> **Note**
>
> La suppression par authentification multifactorielle (MFA) et la gestion des versions S3 ne sont pas prises en charge pour S3 Express One Zone.

AWS Identity and Access Management (IAM) pour S3 Express One Zone

AWS Identity and Access Management (IAM) est un outil Service AWS qui aide les administrateurs à contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (disposer d'autorisations) à utiliser des ressources Amazon S3 dans S3 Express One Zone. Vous pouvez utiliser IAM sans frais supplémentaires.

Par défaut, les utilisateurs ne disposent pas d'autorisations pour les compartiments de répertoires et les opérations S3 Express One Zone. Pour accorder des autorisations d'accès pour les compartiments de répertoires, vous pouvez utiliser IAM pour créer des utilisateurs, des groupes ou des rôles, et attacher des autorisations à ces identités. Pour plus d'informations sur IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Pour fournir l'accès, vous pouvez ajouter des autorisations à vos utilisateurs, groupes ou rôles via les méthodes suivantes :

- Utilisateurs et groupes dans AWS IAM Identity Center : créez un ensemble d'autorisations. Suivez les instructions de la rubrique Création d’un jeu d’autorisations du Guide de l’utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM via un fournisseur d’identité : créez un rôle pour la fédération d’identité. Pour plus d’informations, voir la rubrique Création d’un rôle pour un fournisseur d’identité tiers (fédération) du Guide de l’utilisateur IAM.
- Rôles et utilisateurs IAM : créez un rôle que votre utilisateur peut endosser. Suivez les instructions fournies dans Création d’un rôle pour la délégation d’autorisations à un utilisateur IAM dans le Guide de l’utilisateur IAM.

Par défaut, les compartiments de répertoires sont privés et sont accessibles uniquement par les utilisateurs auxquels l’accès a été explicitement accordé. La limite de contrôle d’accès pour les compartiments de répertoires est définie uniquement au niveau compartiment. En revanche, la limite de contrôle d’accès pour les compartiments à usage général peut être définie au niveau des compartiments, du préfixe ou de la balise d’objet. Cette différence signifie que les compartiments de répertoires sont la seule ressource que vous pouvez inclure dans les politiques de compartiment ou les politiques d’identité IAM pour accéder à S3 Express One Zone.

Avec S3 Express One Zone, outre l’autorisation IAM, vous authentifiez et autorisez les demandes par le biais d’un nouveau mécanisme basé sur les sessions et géré par l’opération d’API `CreateSession`. Vous pouvez utiliser `CreateSession` pour demander des informations d’identification temporaires afin de bénéficier d’un accès à faible latence à votre compartiment. Ces informations d’identification temporaires sont limitées à un compartiment de répertoires spécifique.

Pour travailler avec`CreateSession`, nous vous recommandons d'utiliser la dernière version des AWS SDK ou d'utiliser le AWS Command Line Interface (AWS CLI). Les AWS SDK pris en charge et le gestionnaire AWS CLI gèrent l'établissement, le rafraîchissement et la résiliation des sessions en votre nom.

Vous utilisez des jetons de session avec uniquement des opérations zonales (niveau objet) (à l’exception de `CopyObject` et `HeadBucket`) pour répartir la latence associée à l’autorisation sur un certain nombre de demandes dans une session. Pour les opérations d’API de point de terminaison régional (opérations de niveau compartiment), vous utilisez l’autorisation IAM, qui n’implique pas la gestion d’une session. Pour plus d’informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone et Autorisation `CreateSession`.

Pour plus d’informations sur IAM pour S3 Express One Zone, consultez les rubriques suivantes.

Rubriques

Principaux

Lorsque vous créez une politique basée sur les ressources pour accorder l'accès à vos compartiments, vous devez utiliser l'élément `Principal` pour spécifier la personne ou l'application qui peut effectuer une demande d'action ou d'opération sur cette ressource. Pour des politiques de compartiment de répertoires, vous pouvez utiliser les principaux suivants :

- Un AWS compte
- Un utilisateur IAM
- Un rôle IAM
- Un utilisateur fédéré

Pour plus d'informations, consultez Principal dans le Guide de l'utilisateur IAM.

Ressources

Les Amazon Resource Names (ARN) pour les compartiments de répertoire contiennent l'espace de `s3express` noms Région AWS, l'ID de AWS compte et le nom du compartiment de répertoire, qui inclut l'ID de zone de disponibilité. Pour accéder à votre compartiment de répertoires et y effectuer des actions, vous devez utiliser le format d'ARN suivant :

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--azid--x-s3
```

Pour plus d'informations sur les ARN, consultez Amazon Resource Names (ARNs) dans le Guide de l'utilisateur IAM. Pour plus d'informations sur les ressources, consultez Éléments de politique JSON IAM : Resource dans le Guide de l'utilisateur IAM.

Actions pour S3 Express One Zone

Dans une politique basée sur l'identité IAM ou sur les ressources, vous définissez quelles actions S3 sont autorisées ou refusées. Les actions S3 Express One Zone correspondent à des opérations d'API spécifiques. S3 Express One Zone possède un espace de noms IAM unique distinct de l'espace de noms standard pour Amazon S3. Cet espace de noms est `s3express`.

Lorsque vous accordez l'autorisation `s3express:CreateSession`, cela permet à l'opération d'API `CreateSession` de récupérer les jetons de session lors de l'accès aux opérations d'API de point de terminaison zonal (ou de niveau objet). Ces jetons de session renvoient des informations d'identification qui sont utilisées pour accorder l'accès à toutes les autres opérations d'API de point de terminaison zonal. Par conséquent, il n'est pas nécessaire d'accorder des autorisations d'accès aux opérations d'API zonales en utilisant des politiques IAM. Au lieu de cela, le jeton de session permet l'accès.

Pour plus d'informations sur les opérations d'API de points de terminaison zonaux et régionaux, consultez Mise en réseau pour S3 Express One Zone. Pour en savoir plus sur l'opération d'API `CreateSession`, consultez CreateSession dans la Référence d'API Amazon Simple Storage Service.

Vous pouvez indiquer les actions suivantes dans l'élément `Action` d'une déclaration de politique IAM. Utilisez des politiques pour accorder des autorisations permettant d'effectuer une opération dans AWS. Lorsque vous utilisez une action dans une politique, vous autorisez ou refusez généralement l'accès à l'opération d'API du même nom. Toutefois, dans certains cas, une seule action contrôle l'accès à plusieurs opérations d'API. L'accès aux actions de niveau compartiment peut être accordé uniquement dans des politiques basées sur l'identité IAM (utilisateur ou rôle) et non pas dans des politiques de compartiment.

Actions et clés de condition pour S3 Express One Zone

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:CreateBucket`	`CreateBucket`	Accorde l'autorisation de créer un nouveau compartiment.	Écrire	`s3express:authType` `s3express:LocationName`

Action	API	Description	Niveau d'accès	Clés de condition
				`s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:CreateSession`	`CreateSession`	Accorde l’autorisation de créer un jeton de session, qui est utilisé pour accorder l’accès à toutes les opérations d’API zonales (niveau objet), telles que `PutObject` , `GetObject` , etc.	Écrire	`s3express:authType` `s3express:SessionMode` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:signatureAge` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:DeleteBucket`	`DeleteBucket`	Accorde l’autorisation de supprimer le compartiment nommé dans l’URI.	Écrire	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:DeleteBucketPolicy`	`DeleteBucketPolicy`	Accorde l’autorisation de supprimer la politique sur un compartiment spécifié.	Gestion des autorisations	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:GetBucketPolicy`	`GetBucketPolicy`	Accorde l’autorisation de renvoyer la politique du compartiment spécifié.	Lecture	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:ListAllMyDirectoryBuckets`	`ListDirectoryBuckets`	Accorde l’autorisation de répertorier tous les compartiments de répertoires appartenant à l’expéditeur authentifié de la demande.	Liste	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Action	API	Description	Niveau d'accès	Clés de condition
`s3express:PutBucketPolicy`	`PutBucketPolicy`	Accorde l'autorisation d'ajouter ou de remplacer une politique de compartiment sur un compartiment.	Gestion des autorisations	`s3express:authType` `s3express:ResourceAccount` `s3express:signatureversion` `s3express:TlsVersion` `s3express:x-amz-content-sha256`

Clés de condition pour S3 Express One Zone

S3 Express One Zone définit les clés de condition suivantes que vous pouvez utiliser dans l'élément `Condition` d'une politique IAM. Vous pouvez utiliser ces clés pour affiner les conditions d'application de la déclaration de politique.

Clé de condition	Description	Type
`s3express:authType`	Filtre l'accès en fonction de la méthode d'authentification. Pour limiter les requêtes entrantes afin d'utiliser d'une méthode d'authentification spécifique, vous pouvez utiliser cette clé de condition facultative. Par exemple, vous pouvez utiliser cette clé de condition pour	Chaîne

Clé de condition	Description	Type
	autoriser uniquement l'en-tête HTTP`Authorization` pour l'authentification de la demande. Valeurs valides : `REST-HEADER` , `REST-QUERY-STRING`	
`s3express:LocationName`	Filtre l’accès à l’opération d’API `CreateBucket` en fonction d’un ID de zone de disponibilité (ID d’AZ) spécifique, par exemple, `usw2-az1`. Exemple de valeur : `usw2-az1`	Chaîne
`s3express:ResourceAccount`	Filtre l'accès en fonction de l' Compte AWS identifiant du propriétaire de la ressource. Pour restreindre l'accès des utilisateurs, des rôles ou des applications aux compartiments de répertoire appartenant à un Compte AWS ID spécifique, vous pouvez utiliser la clé de `s3express:ResourceAccount` condition `aws:ResourceAccount` ou. Vous pouvez utiliser cette clé de condition dans les politiques d'identité AWS Identity and Access Management (IAM) ou dans les politiques de point de terminaison du cloud privé virtuel (VPC). Par exemple, vous pouvez utiliser cette clé de condition pour empêcher les clients de votre VPC d'accéder à des buckets dont vous n'êtes pas le propriétaire. Exemple de valeur : `111122223333`	Chaîne

Clé de condition	Description	Type
`s3express:SessionMode`	Filtre l’accès en fonction de l’autorisation demandée par l’opération d’API `CreateSession` . Par défaut, la session est `ReadWrite` . Vous pouvez utiliser cette clé de condition pour limiter l’accès à `ReadOnly` ou pour refuser explicitement l’accès `ReadWrite` . Pour plus d'informations, consultez Exemples de politiques de compartiment de répertoires pour S3 Express One Zone et CreateSession dans la référence de l'API Amazon Simple Storage Service. Valeurs valides : `ReadWrite` , `ReadOnly`	Chaîne
`s3express:signatureAge`	Filtre l’accès en fonction de l’âge en millisecondes de la signature de la demande. Cette condition fonctionne uniquement pour les URL présignées. Dans AWS la version 4 de signature, la clé de signature est valide jusqu'à sept jours. Par conséquent, les signatures ne restent valides que pendant sept jours. Pour plus d’informations, consultez Introduction à la signature des demandes dans la Référence d’API Amazon Simple Storage Service. Vous pouvez utiliser cette condition pour limiter davantage la durée de la signature. Exemple de valeur : `600000`	Numérique

Clé de condition	Description	Type
`s3express:signaturversion`	Identifie la version de AWS Signature que vous souhaitez prendre en charge pour les demandes authentifiées. Pour les demandes authentifiées, S3 Express One Zone prend en charge Signature Version 4. Valeur valide : `"AWS4-HMAC-SHA256"` (identifie la version 4 de la signature)	Chaîne
`s3express:TlsVersion`	Filtre l'accès en fonction de la version TLS utilisée par le client. Vous pouvez utiliser la clé de `s3:TlsVersion` condition pour écrire des politiques IAM, de point de terminaison de cloud privé virtuel (VPCE) ou de bucket qui limitent l'accès des utilisateurs ou des applications aux compartiments d'annuaire en fonction de la version TLS utilisée par le client. Vous pouvez utiliser cette clé de condition pour écrire des politiques qui nécessitent une version TLS minimale. Exemple de valeur : `1.3`	Numérique

Clé de condition	Description	Type
`s3express:x-amz-content-sha256`	Filtre l'accès en fonction du contenu non signé dans votre compartiment. Vous pouvez utiliser cette clé de condition pour interdire les contenus non signés dans votre compartiment. Lorsque vous utilisez Signature Version 4, pour les demandes qui utilisent l'en-tête `Authorization` , vous ajoutez l'en-tête `x-amz-content-sha256` dans le calcul de signature, puis définissez sa valeur sur la charge utile du hachage. Vous pouvez utiliser cette clé de condition dans votre politique de compartiment pour refuser tous les chargements où les charges utiles ne sont pas signées. Par exemple : • Refuser les chargements qui utilisent l'en-tête `Authorization` pour authentifier les requêtes mais ne signent pas la charge utile. Pour plus d'informations, consultez Transfert de la charge utile en un seul fragment dans la Référence d'API Amazon Simple Storage Service. • Refusez les téléchargements utilisant des URL présignées. Les URL présignées ont toujours une `UNSIGNED_PAYLOAD` . Pour plus d'informations, consultez Authentification des demandes et Méthodes d'authentification dans la Référence d'API Amazon Simple Storage Service. Valeur valide : `UNSIGNED-PAYLOAD`	Chaîne

Comment les opérations d'API sont autorisées et authentifiées

Le tableau suivant répertorie les informations d'autorisation et d'authentification pour les opérations d'API S3 Express One Zone. Pour chaque opération d'API, le tableau indique le nom de l'opération d'API, l'action IAM, le type de point de terminaison (régional ou zonal) et le mécanisme d'autorisation (IAM ou basé sur une session). Ce tableau indique également les endroits où l'accès intercompte est pris en charge. L'accès aux actions de niveau compartiment peut être accordé uniquement dans des politiques basées sur l'identité IAM (utilisateur ou rôle), et non pas dans des politiques de compartiment.

API	Type de point de terminaison	Action IAM	Accès intercomptes
`CreateBucket`	Régional	`s3express:CreateBucket`	Non
`DeleteBucket`	Régional	`s3express:DeleteBucket`	Non
`ListDirectoryBuckets`	Régional	`s3express:ListAllMyDirectoryBuckets`	Non
`PutBucketPolicy`	Régional	`s3express:PutBucketPolicy`	Non
`GetBucketPolicy`	Régional	`s3express:GetBucketPolicy`	Non
`DeleteBucketPolicy`	Régional	`s3express:DeleteBucketPolicy`	Non
`CreateSession`	Zonal	`s3express:CreateSession`	Oui
`CopyObject`	Zonal	`s3express:CreateSession`	Oui
`DeleteObject`	Zonal	`s3express:CreateSession`	Oui
`DeleteObjects`	Zonal	`s3express:CreateSession`	Oui
`HeadObject`	Zonal	`s3express:CreateSession`	Oui

API	Type de point de terminaison	Action IAM	Accès intercomptes
`PutObject`	Zonal	`s3express:CreateSession`	Oui
`GetObjectAttributes`	Zonal	`s3express:CreateSession`	Oui
`ListObjectsV2`	Zonal	`s3express:CreateSession`	Oui
`HeadBucket`	Zonal	`s3express:CreateSession`	Oui
`CreateMultipartUpload`	Zonal	`s3express:CreateSession`	Oui
`UploadPart`	Zonal	`s3express:CreateSession`	Oui
`UploadPartCopy`	Zonal	`s3express:CreateSession`	Oui
`CompleteMultipartUpload`	Zonal	`s3express:CreateSession`	Oui
`AbortMultipartUpload`	Zonal	`s3express:CreateSession`	Oui
`ListParts`	Zonal	`s3express:CreateSession`	Oui
`ListMultipartUploads`	Zonal	`s3express:CreateSession`	Oui

Politiques basées sur l’identité IAM pour S3 Express One Zone

Avant de créer des compartiments de répertoires ou d’utiliser la classe de stockage Amazon S3 Express One Zone, vous devez accorder les autorisations nécessaires à votre rôle ou à vos utilisateurs AWS Identity and Access Management (IAM). Cet exemple de politique autorise l’accès à l’opération d’API `CreateSession` (à utiliser avec les opérations d’API de point de terminaison

zonal (niveau objet)) et à toutes les opérations d'API de point de terminaison régional (niveau compartiment). Cette politique autorise l'utilisation de l'opération d'API `CreateSession` avec tous les compartiments de répertoires, mais les opérations d'API de point de terminaison régional sont autorisées à être utilisées uniquement avec le compartiment de répertoires spécifié. Pour utiliser cet exemple de politique, remplacez *`user input placeholders`* par vos propres informations.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "AllowAccessRegionalEndpointAPIs",
             "Effect": "Allow",
             "Action": [
                 "s3express:DeleteBucket",
                 "s3express:DeleteBucketPolicy",
                 "s3express:CreateBucket",
                 "s3express:PutBucketPolicy",
                 "s3express:GetBucketPolicy",
                 "s3express:ListAllMyDirectoryBuckets"
             ],

             "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--azid--x-s3/*"
         },
         {
             "Sid": "AllowCreateSession",
             "Effect": "Allow",
             "Action": "s3express:CreateSession",
             "Resource": "*"
         }
     ]
 }
```

Exemples de politiques de compartiment de répertoires pour S3 Express One Zone

Cette section fournit des exemples de politiques de compartiment de répertoires à utiliser avec la classe de stockage Amazon S3 Express One Zone. Pour utiliser ces politiques, remplacez les *`user input placeholders`* par vos propres informations.

L'exemple de politique de compartiment suivant permet au Compte AWS d'ID *111122223333* d'utiliser l'opération d'API `CreateSession` avec la session `ReadWrite` par défaut pour le compartiment de répertoires spécifié. Cette politique accorde l'accès aux opérations d'API de point de terminaison zonal (niveau objet).

Example – Politique de compartiment permettant d'autoriser les appels **`CreateSession`** avec la session **`ReadWrite`** par défaut

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccess",
            "Effect": "Allow",
            "Resource": "arn:aws:s3express:us-west-2:account-id:bucket/bucket-base-name--azid--x-s3",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            },
            "Action": [
                "s3express:CreateSession"
            ]
        }
    ]
}
```

Example – Politique de compartiment permettant d'autoriser les appels **`CreateSession`** avec une session **`ReadOnly`**

L'exemple de politique de compartiment suivant permet au Compte AWS d'ID *111122223333* d'utiliser l'opération d'API `CreateSession`. Cette politique utilise la clé de condition `s3express:SessionMode` avec la valeur `ReadOnly` pour définir une session en lecture seule.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express:CreateSession",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3express:SessionMode": "ReadOnly"
                }
            }
        }
    ]
}
```

Example – Politique de compartiment permettant d'autoriser l'accès intercompte pour les appels **`CreateSession`**

L'exemple de politique de compartiment suivant permet au Compte AWS d'ID *111122223333* d'utiliser l'opération d'API `CreateSession` avec le compartiment de répertoires spécifié détenu par le Compte AWS d'ID *444455556666*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccount",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "s3express:CreateSession"
            ],
            "Resource": "arn:aws:s3express:us-west-2:444455556666:bucket/bucket-base-name--azid--x-s3"

        }
    ]
}
```

Autorisation `CreateSession`

Amazon S3 Express One Zone prend en charge à la fois l'autorisation AWS Identity and Access Management (AWS IAM) et l'autorisation basée sur les sessions :

- Pour utiliser les opérations d'API de point de terminaison régional (opérations au niveau du compartiment ou du plan de contrôle) avec S3 Express One Zone, vous utilisez le modèle d'autorisation IAM, qui n'implique pas de gestion de session. Les autorisations sont accordées pour les actions, de façon individuelle. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.
- Pour utiliser les opérations d'API de point de terminaison zonal (opérations de niveau objet ou plan de données), vous utilisez l'opération d'API `CreateSession` pour créer et gérer des sessions optimisées afin d'autoriser les demandes de données à faible latence. Pour récupérer et utiliser un jeton de session, vous devez autoriser l'action `s3express:CreateSession` pour votre compartiment de répertoires dans une politique basée sur l'identité ou une politique de compartiment. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone. Si vous accédez à S3 Express One Zone dans la console Amazon S3, via l'AWS Command Line Interface (AWS CLI) ou à l'aide des kits AWS SDK, S3 Express One Zone crée une session en votre nom.

Si vous utilisez l'API REST Amazon S3, vous pouvez ensuite utiliser l'opération d'API `CreateSession` pour obtenir des informations d'identification de sécurité temporaires qui incluent un ID de clé d'accès, une clé d'accès secrète, un jeton de session et une date d'expiration. Les informations d'identification temporaires fournissent les mêmes autorisations que les informations d'identification de sécurité à long terme, telles que les informations d'identification d'utilisateur IAM, mais les informations d'identification de sécurité temporaires doivent inclure un jeton de session.

Mode session

Le mode session définit l'étendue de la session. Dans votre politique de compartiment, vous pouvez spécifier la clé de condition `s3express:SessionMode` pour contrôler qui peut créer une session `ReadWrite` ou `ReadOnly`. Pour plus d'informations sur les sessions `ReadWrite` et `ReadOnly`, consultez le paramètre `x-amz-create-session-mode` pour CreateSession dans la Référence d'API Amazon S3. Pour plus d'informations sur la politique de compartiment à créer, consultez Exemples de politiques de compartiment de répertoires pour S3 Express One Zone.

Jeton de session

Lorsque vous effectuez un appel en utilisant des informations d'identification de sécurité temporaires, l'appel doit inclure un jeton de session. Le jeton de session est renvoyé avec les informations d'identification temporaires. Un jeton de session est limité à votre compartiment de répertoires et est utilisé pour vérifier que les informations d'identification de sécurité sont valides et n'ont pas expiré. Pour protéger vos sessions, les informations d'identification de sécurité temporaires expirent au bout de 5 minutes.

`CopyObject` et `HeadBucket`

Les informations d'identification de sécurité temporaires sont limitées à un compartiment de répertoires spécifique et sont automatiquement activées pour tous les appels d'API d'opérations zonales (niveau objet) vers un compartiment de répertoires donné. Contrairement aux autres opérations d'API de point de terminaison zonal, `CopyObject` et `HeadBucket` n'utilisent pas l'authentification `CreateSession`. Toutes les demandes `CopyObject` et `HeadBucket` doivent être authentifiées et signées au moyen d'informations d'identification IAM. Toutefois, `CopyObject` et `HeadBucket` sont toujours autorisées par `s3express:CreateSession`, comme les autres opérations d'API de point de terminaison zonal.

Pour plus d'informations, veuillez consulter CreateSession dans la Référence d'API Amazon Simple Storage Service.

Bonnes pratiques de sécurité pour S3 Express One Zone

Amazon S3 Express One Zone fournit différentes fonctionnalités de sécurité à prendre en compte lorsque vous développez et implémentez vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des recommandations utiles plutôt que comme des prescriptions.

Paramètres de blocage d'accès public définis par défaut et de propriété des objets

Pour utiliser la classe de stockage S3 Express One Zone, vous devez utiliser un compartiment de répertoires S3. Les compartiments de répertoires prennent en charge le blocage de l'accès public S3 et la propriété des objets S3. Ces fonctionnalités S3 sont utilisées pour auditer et gérer l'accès à vos compartiments et objets.

Par défaut, tous les paramètres de blocage de l'accès public des compartiments de répertoires sont activés. En outre, la propriété des objets est définie sur Propriétaire du compartiment appliqué, ce qui signifie que les listes de contrôle d'accès (ACL) sont désactivées. Ces paramètres ne peuvent pas être modifiés. Pour plus d'informations sur ces fonctionnalités, consultez the section called "Blocage de l'accès public" et the section called "Contrôle de la propriété des objets".

> **Note**
>
> Vous ne pouvez pas autoriser l'accès aux objets stockés dans des compartiments de répertoires. Vous pouvez accorder l'accès uniquement à vos compartiments de répertoires. Le modèle d'autorisation pour S3 Express One Zone est différent du modèle d'autorisation pour Amazon S3. Pour plus d'informations, consultez Autorisation `CreateSession`.

Authentification et autorisation

Les mécanismes d'authentification et d'autorisation pour S3 Express One Zone diffèrent selon que vous envoyez des demandes aux opérations d'API de point de terminaison zonal ou aux opérations d'API de point de terminaison régional. Les opérations d'API zonales sont des opérations de niveau objet (plan de données). Les opérations d'API régionales sont des opérations de niveau compartiment (plan de contrôle).

Avec S3 Express One Zone, vous authentifiez et autorisez les demandes adressées aux opérations d'API de point de terminaison zonal par le biais d'un nouveau mécanisme basé sur les sessions, optimisé pour fournir une latence minimale. Avec l'authentification basée sur les sessions, les kits AWS SDK utilisent l'opération d'API `CreateSession` pour demander des informations d'identification temporaires qui fournissent un accès à faible latence à votre compartiment de répertoires. Ces informations d'identification temporaires sont limitées à un compartiment de répertoires spécifique et expirent au bout de 5 minutes. Vous pouvez utiliser ces informations d'identification temporaires pour signer des appels d'API zonaux (niveau objet). Pour plus d'informations, consultez Autorisation `CreateSession`.

Signature de demandes à l'aide des informations d'identification S3 Express One Zone

Vous utilisez vos informations d'identification S3 Express One Zone pour signer les demandes d'API de point de terminaison zonal (niveau objet) avec AWS Signature Version 4, avec `s3express` comme nom de service. Lorsque vous signez vos demandes, utilisez la clé secrète renvoyée par `CreateSession` et fournissez également le jeton de session avec `x-amzn-s3session-token` `header`. Pour plus d'informations, consultez CreateSession.

Les kits AWS SDK pris en charge pour la classe S3 Express One Zone gèrent les informations d'identification et la signature en votre nom. Nous vous recommandons d'utiliser les kits AWS SDK pour S3 Express One Zone afin d'actualiser les informations d'identification et de signer les demandes pour vous.

Signature de demandes avec des informations d'identification IAM

Tous les appels d'API régionaux (niveau compartiment) doivent être authentifiés et signés par des informations d'identification AWS Identity and Access Management (IAM) plutôt que par des informations d'identification de session temporaires. Les informations d'identification IAM comprennent l'ID de clé d'accès et la clé d'accès secrète des identités IAM. Toutes les demandes `CopyObject` et `HeadBucket` doivent également être authentifiées et signées au moyen d'informations d'identification IAM.

Pour réduire au maximum la latence de vos appels d'opérations zonales (niveau objet), nous vous recommandons d'utiliser les informations d'identification S3 Express One Zone obtenues lors de l'appel de `CreateSession` pour signer vos demandes, à l'exception des demandes adressées à `CopyObject` et `HeadBucket`.

Utiliser AWS CloudTrail

AWS CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS dans Amazon S3. Vous pouvez utiliser les informations collectées par CloudTrail pour déterminer les éléments suivants :

- La demande qui a été adressée à Amazon S3
- L'adresse IP à partir de laquelle la demande a été effectuée
- La personne ayant effectué la demande
- La date et l'heure où la demande a été effectuée
- Des détails supplémentaires sur la demande

Lorsque vous configurez votreCompte AWS, CloudTrail est activé par défaut. Les opérations d'API de point de terminaison régional suivantes (opérations d'API au niveau du compartiment ou du plan de contrôle, opérations d'API) sont enregistrées. CloudTrail

- `CreateBucket`
- `DeleteBucket`

- `DeleteBucketPolicy`
- `PutBucketPolicy`
- `GetBucketPolicy`
- `ListDirectoryBuckets`

Vous pouvez consulter les événements récents dans la CloudTrail console. Pour créer un enregistrement continu de l'activité et des événements de vos compartiments Amazon S3, vous pouvez créer un suivi dans la CloudTrail console. Pour plus d'informations, consultez Creating a trail dans le Guide de l'utilisateur AWS CloudTrail.

> **Note**
>
> Pour S3 Express One Zone, la CloudTrail journalisation des opérations d'API du point de terminaison zonal (au niveau de l'objet ou du plan de données) (par exemple, `PutObject` ou`GetObject`) n'est pas prise en charge.

Mise en œuvre de la surveillance à l'aide des outils de surveillance AWS

La surveillance joue en rôle important dans le maintien de la fiabilité, de la sécurité, de la disponibilité et des performances d'Amazon S3 et de vos solutions AWS. AWS fournit différents outils et services pour vous aider à surveiller Amazon S3 et vos autres Services AWS. Par exemple, vous pouvez surveiller CloudWatch les métriques Amazon pour Amazon S3, en particulier les métriques `NumberOfObjects` de stockage `BucketSizeBytes` et.

Les objets stockés dans la classe de stockage S3 Express One Zone ne seront pas reflétés dans les métriques de stockage `BucketSizeBytes` et `NumberOfObjects` d'Amazon S3. Toutefois, les métriques de stockage `BucketSizeBytes` et `NumberOfObjects` sont prises en charge pour S3 Express One Zone. Pour consulter les métriques de votre choix, vous pouvez différencier les classes de stockage Amazon S3 de la classe de stockage S3 Express One Zone en spécifiant une dimension `StorageType`. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.

Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch et Surveillance d'Amazon S3.

Optimisation des performances d’Amazon S3 Express One Zone

Amazon S3 Express One Zone est une classe de stockage S3 à zone de disponibilité (AZ) unique, à hautes performances, spécialement conçue pour fournir un accès aux données constant en moins de dix millisecondes pour vos applications les plus sensibles à la latence. S3 Express One Zone est la première classe de stockage S3 qui vous permet de regrouper des ressources AWS de calcul et de stockage d’objets hautes performances, telles qu’Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service et Amazon Elastic Container Service, au sein d’une même zone de disponibilité. Le regroupement des ressources de stockage et de calcul optimise les performances et les coûts de calcul et augmente la vitesse de traitement des données.

S3 Express One Zone offre une élasticité des performances similaire à celle des autres classes de stockage S3, mais avec des latences du premier octet constantes de demande de lecture et d’écriture en moins de dix millisecondes, jusqu’à 10 fois plus rapides que S3 Standard. S3 Express One Zone est entièrement conçu pour prendre en charge un débit en rafale jusqu’à des niveaux d’agrégation très élevés. La classe de stockage S3 Express One Zone utilise une architecture personnalisée pour optimiser les performances et garantir une faible latence des demandes en stockant les données sur du matériel à hautes performances. Le protocole d’objet pour S3 Express One Zone a été amélioré afin de rationaliser l’authentification et la surcharge de métadonnées.

Pour augmenter encore la vitesse d’accès et prendre en charge des centaines de milliers de demandes par seconde, S3 Express One Zone stocke les données dans un nouveau type de compartiment : un compartiment de répertoires Amazon S3. Chaque compartiment de répertoires S3 peut prendre en charge des centaines de milliers de transactions par seconde (TPS).

La combinaison d’un matériel et de logiciels spécialement conçus à hautes performances, offrant une vitesse d’accès aux données de l’ordre de quelques millisecondes, et de compartiments de répertoires pouvant être mis à l’échelle pour un grand nombre de transactions par seconde fait de S3 Express One Zone la meilleure classe de stockage Amazon S3 pour les opérations exigeantes en termes de demandes ou les applications critiques en matière de performances.

Les rubriques suivantes décrivent les bonnes pratiques à suivre et les modèles de conception pour optimiser les performances des applications qui utilisent la classe de stockage S3 Express One Zone.

Rubriques

Directives d’optimisation des performances et modèles de conception pour S3 Express One Zone

Lors du développement d’applications qui chargent et récupèrent les objets depuis Amazon S3 Express One Zone, suivez nos bonnes pratiques pour optimiser les performances. Pour utiliser la classe de stockage S3 Express One Zone, vous devez créer un compartiment de répertoires S3. La classe de stockage S3 Express One Zone n’est pas prise en charge pour une utilisation avec les compartiments S3 à usage général.

Pour accéder aux directives d’optimisation des performances relatives à toutes les autres classes de stockage Amazon S3 et aux compartiments à usage général S3, consultez Schémas de conception des bonnes pratiques : optimisation des performances Amazon S3.

Pour optimiser les performances pour votre application lorsque vous utilisez la classe de stockage et les compartiments de répertoires S3 Express One Zone, nous recommandons les directives et modèles de conception suivants.

Rubriques

Regroupement du stockage S3 Express One Zone et de vos ressources de calcul AWS

Chaque compartiment de répertoires est stocké dans une zone de disponibilité unique que vous sélectionnez lorsque vous créez le compartiment. Vous pouvez commencer par créer un nouveau compartiment de répertoires dans une zone de disponibilité locale pour vos charges de travail ou vos ressources de calcul. Vous pouvez alors commencer immédiatement des opérations de lecture et d’écriture à très faible latence. Les compartiments de répertoires sont les premiers compartiments S3 dans lesquels vous pouvez choisir la zone de disponibilité dans une Région AWS, afin de réduire la latence entre le calcul et le stockage.

Si vous accédez à des compartiments de répertoires dans différentes zones de disponibilité, la latence augmente. Pour optimiser les performances, nous vous recommandons d'accéder à un compartiment de répertoires depuis des instances Amazon Elastic Container Service, Amazon Elastic Kubernetes Service et Amazon Elastic Compute Cloud situées dans la même zone de disponibilité, quand cela est possible.

Compartiments de répertoires

Chaque compartiment de répertoires peut prendre en charge des centaines de milliers de transactions par seconde (TPS). Contrairement aux compartiments à usage général, les compartiments de répertoires organisent les clés de manière hiérarchique dans des répertoires plutôt qu'avec des préfixes. Un préfixe est une chaîne de caractères au début du nom de la clé d'objet. Vous pouvez voir les préfixes comme un moyen d'organiser vos données de la même manière que les répertoires. Toutefois, les préfixes ne sont pas des répertoires.

Les préfixes organisent les données dans un espace de noms plat au sein de compartiments à usage général, et le nombre de préfixes dans un compartiment à usage général est illimité. Chaque préfixe peut atteindre au moins 3 500 `HEAD` requêtes`PUT`/`POST`/`DELETE`ou 5 500`GET`/par seconde. Vous pouvez également mettre en parallèle les demandes sur plusieurs préfixes pour mettre à l'échelle les performances. Toutefois, dans le cas d'opérations de lecture et d'écriture, cette mise à l'échelle se fait progressivement et n'est pas instantanée. Pendant la mise à l'échelle des compartiments à usage général pour atteindre votre nouveau taux de demandes supérieur, vous pouvez recevoir des erreurs de code d'état HTTP 503 (Service non disponible).

Avec un espace de noms hiérarchique, le délimiteur figurant dans la clé d'objet est important. Le seul délimiteur pris en charge est la barre oblique (`/`). Les répertoires sont déterminés par les limites des délimiteurs. Par exemple, la clé d'objet `dir1/dir2/file1.txt` entraîne la création automatique des répertoires `dir1/` et `dir2/`, et l'ajout de l'objet `file1.txt` dans le répertoire `/dir2`, dans le chemin `dir1/dir2/file1.txt`.

Les répertoires créés lorsque les objets sont chargés dans des compartiments de répertoires ne sont soumis à aucune limite TPS par préfixe et sont automatiquement prédimensionnés pour réduire la probabilité d'erreurs HTTP 503 (Service non disponible). Cette mise à l'échelle automatique permet à vos applications de mettre en parallèle les demandes de lecture et d'écriture dans et entre les répertoires, selon les besoins.

Mise en parallèle des demandes de mise à l'échelle horizontale des compartiments de répertoires

Vous pouvez optimiser les performances en adressant plusieurs demandes simultanées aux compartiments de répertoires pour répartir vos demandes sur des connexions distinctes et augmenter au maximum la bande passante accessible. S3 Express One Zone n'a aucune limite quant au nombre de connexions établies avec votre compartiment de répertoires. Les répertoires individuels peuvent mettre à l'échelle horizontalement et automatiquement les performances quand un grand nombre d'écritures simultanées sont effectuées dans le même répertoire.

Lorsqu'une clé d'objet est initialement créée et que son nom de clé inclut un répertoire, le répertoire est automatiquement créé pour l'objet. Les chargements d'objets ultérieurs vers ce même répertoire ne nécessitent pas la création du répertoire, ce qui réduit la latence lors des chargements d'objets vers les répertoires existants.

Des structures de répertoires superficielles et profondes sont prises en charge pour stocker des objets dans un compartiment de répertoires, mais les compartiments de répertoires sont automatiquement mis à l'échelle horizontalement, avec une latence plus faible lors de chargements simultanés vers le même répertoire ou vers des répertoires similaires parallèles.

Utilisation de l'authentification basée sur les sessions

S3 Express One Zone et les compartiments de répertoires prennent en charge un nouveau mécanisme d'autorisation basé sur les sessions pour authentifier et autoriser les demandes adressées à un compartiment de répertoires. Avec l'authentification basée sur les sessions, les kits AWS SDK utilisent automatiquement l'opération d'API `CreateSession` pour créer un jeton de session temporaire utilisable pour l'autorisation à faible latence des demandes de données adressées à un compartiment de répertoires.

Les kits AWS SDK utilisent l'opération d'API `CreateSession` pour demander des informations d'identification temporaires, puis créent et actualisent automatiquement des jetons pour vous, en votre nom, toutes les 5 minutes. Pour tirer parti des avantages en matière de performances de la classe de stockage S3 Express One Zone, nous vous recommandons d'utiliser les kits AWS SDK pour lancer et gérer la demande d'API `CreateSession`. Pour plus d'informations sur ce modèle basé sur les sessions, consultez Autorisation `CreateSession`.

Bonnes pratiques supplémentaires en matière de somme de contrôle S3

S3 Express One Zone vous offre la possibilité de choisir l'algorithme de somme de contrôle utilisé pour valider vos données pendant le chargement ou le téléchargement. Vous pouvez sélectionner

l’un des algorithmes de contrôle d’intégrité des données Secure Hash Algorithms (SHA) ou Cyclic Redundancy Check (CRC) suivants : CRC32, CRC32C, SHA-1 ou SHA-256. Les checksums basés sur MD5 ne sont pas pris en charge avec la classe de stockage S3 Express One Zone.

CRC32 est la somme de contrôle par défaut utilisée par les kits AWS SDK lors de la transmission de données vers ou depuis S3 Express One Zone. Nous recommandons d’utiliser CRC32 et CRC32C pour optimiser les performances avec la classe de stockage S3 Express One Zone.

Utilisation de la dernière version des kits AWS SDK et des bibliothèques CRT (Common Runtime)

Plusieurs des kits AWS SDK fournissent également les bibliothèques AWS CRT (Common Runtime) pour accélérer encore les performances des clients S3. Ces kits SDK incluent le kit AWS SDK for Java 2.x, le kit AWS SDK for C++ et le kit AWS SDK for Python (Boto3). Le client S3 basé sur CRT transfère des objets vers et depuis S3 Express One Zone avec des performances et une fiabilité améliorées en utilisant automatiquement l’opération d’API de chargement partitionné et les extractions de plages d’octets pour automatiser la mise à l’échelle horizontale des connexions.

Pour obtenir les meilleures performances avec la classe de stockage S3 Express One Zone, nous vous recommandons d’utiliser la dernière version des kits AWS SDK qui incluent les bibliothèques CRT ou d’utiliser l’AWS Command Line Interface (AWS CLI).

Dépannage des performances

Nouvelle tentative de demandes pour les applications sensibles à la latence

La classe S3 Express One Zone est spécialement conçue pour fournir des niveaux élevés constants de performances sans réglages supplémentaires. Toutefois, la définition de valeurs de délai d’attente et de nouvelles tentatives agressives contribue également à garantir une latence et des performances constantes. Les kits SDK AWS ont des valeurs configurables de délai d'expiration et de nouvelle tentatives que vous pouvez adapter aux tolérances de votre application spécifique

Association des types d’instances Amazon EC2 et des bibliothèques AWS CRT (Common Runtime)

Les applications qui effectuent un grand nombre d’opérations de lecture et d’écriture ont probablement besoin de plus de mémoire et de capacité de calcul que les autres. Lorsque vous lancez vos instances Amazon Elastic Compute Cloud (Amazon EC2) pour votre charge de travail exigeante en performances, choisissez les types d’instances qui disposent de la quantité de ces ressources dont votre application a besoin. Le stockage hautes performances S3 Express One Zone est idéalement associé à des types d’instances plus grands et plus récents, dotés d’une plus grande

quantité de mémoire système et de CPU et GPU plus puissants, qui peuvent tirer parti d'un stockage plus performant. Nous vous recommandons également d'utiliser les dernières versions des kits AWS SDK compatibles CRT, qui accélèrent encore les demandes de lecture et d'écriture en parallèle.

Utilisation d'une authentification basée sur les sessions dans les kits AWS SDK à la place des API REST HTTP

Avec Amazon S3, vous pouvez également optimiser les performances lorsque vous utilisez des demandes d'API REST HTTP en suivant les mêmes bonnes pratiques que celles incluses dans les kits AWS SDK. Toutefois, avec le mécanisme d'autorisation et d'authentification basé sur les sessions utilisé par S3 Express One Zone, nous vous recommandons vivement d'utiliser les kits AWS SDK pour gérer `CreateSession` et son jeton de session géré. Les kits AWS SDK créent et actualisent automatiquement les jetons en votre nom à l'aide de l'opération d'API `CreateSession`. L'utilisation de `CreateSession` économise du temps de latence aller-retour par demande vers AWS Identity and Access Management (IAM) pour autoriser chaque demande.

Développement avec S3 Express One Zone

Amazon S3 Express One Zone est la première classe de stockage S3 dans laquelle vous pouvez sélectionner une zone de disponibilité unique avec la possibilité de regrouper le stockage d'objets et les ressources de calcul, ce qui assure la vitesse d'accès la plus élevée possible. Avec la classe de stockage S3 Express One Zone, vous utilisez des compartiments de répertoires S3 pour stocker vos données. Chaque compartiment de répertoires utilise la classe de stockage S3 Express One Zone pour stocker les objets dans une zone de disponibilité unique que vous pouvez sélectionner lors de la création du compartiment.

Une fois que vous avez créé votre compartiment de répertoires, vous pouvez immédiatement commencer des opérations de lecture et d'écriture à très faible latence. Vous pouvez communiquer avec votre compartiment de répertoires à l'aide d'une connexion de point de terminaison via un cloud privé virtuel (VPC), ou vous pouvez utiliser les opérations d'API zonales et régionales pour gérer vos objets et compartiments de répertoires. Vous pouvez également utiliser la classe de stockage S3 Express One Zone via la console Amazon S3, l'AWS Command Line Interface (AWS CLI), les kits AWS SDK et l'API REST Amazon S3.

La classe de stockage Amazon S3 Express One Zone est conçue pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenue par le contrat de niveau de service Amazon S3. Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu

pour gérer les défaillances simultanées de périphériques en détectant et réparant rapidement toute perte de redondance. En cas de panne de l’appareil existant, S3 Express One Zone transfère automatiquement les demandes vers de nouveaux appareils au sein d’une zone de disponibilité. Cette redondance permet de garantir un accès ininterrompu à vos données au sein d’une zone de disponibilité.

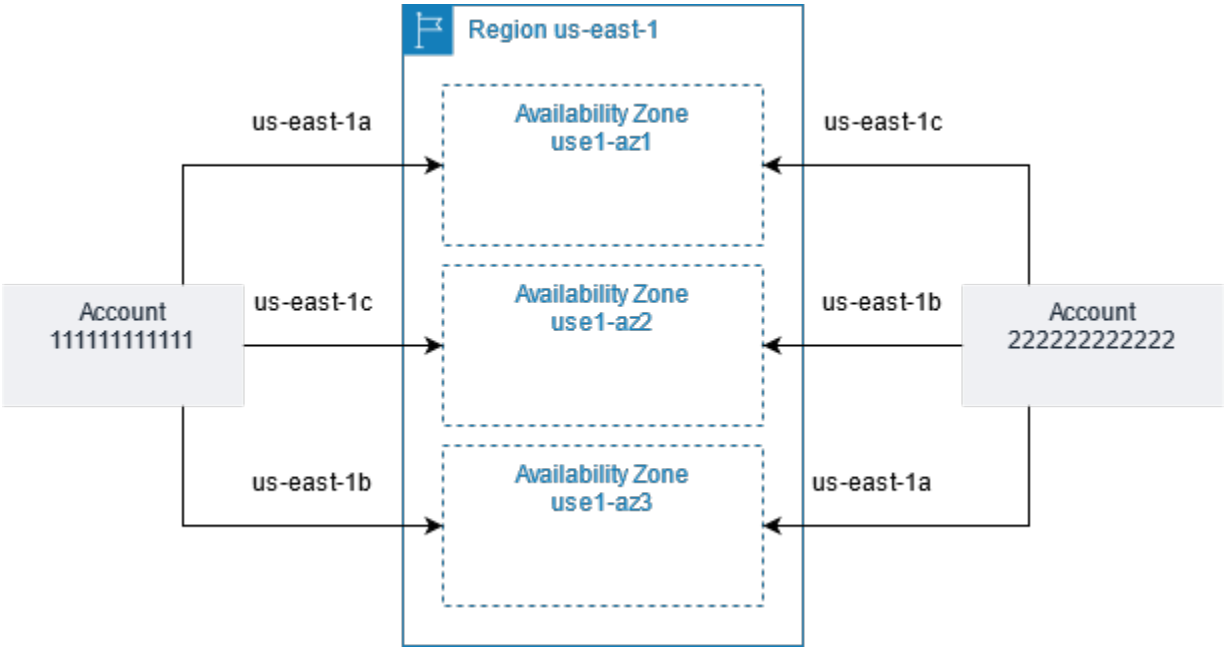
Rubriques

Régions et zones de disponibilité S3 Express One Zone

Une zone de disponibilité est un ou plusieurs centres de données discrets dotés d'une alimentation, d'un réseau et d'une connectivité redondants dans une Région AWS. Pour optimiser les récupérations à faible latence, les objets de la classe de stockage Amazon S3 Express One Zone sont stockés de manière redondante dans des compartiments de répertoires S3 dans une zone de disponibilité unique locale à votre charge de travail de calcul. Lorsque vous créez un bucket d'annuaire, vous choisissez la zone de disponibilité et Région AWS l'emplacement de votre bucket.

AWS fait correspondre les zones de disponibilité physiques de manière aléatoire aux noms des zones de disponibilité de chacune d'entre elles Compte AWS. Cette approche permet de répartir les ressources entre les zones de disponibilité au lieu de concentrer les ressources dans la première zone de disponibilité de chaque région. Région AWS Par conséquent, il est Compte AWS possible que la zone `us-east-1a` de disponibilité de votre site ne représente pas le même emplacement physique que celle `us-east-1a` d'une autre Compte AWS. Pour plus d'informations, consultez la section Régions et zones de disponibilité dans le guide de l'utilisateur Amazon EC2.

Pour coordonner les zones de disponibilité entre les comptes, vous devez utiliser un ID de zone de disponibilité, qui représente l’identifiant unique et cohérent d’une zone de disponibilité. Par exemple, `use1-az1` il s'agit d'un identifiant AZ pour la `us-east-1` région et il a le même emplacement physique dans chaque région Compte AWS. L’illustration suivante montre comment les ID de zone de disponibilité sont les mêmes pour tous les comptes, même si les noms des zones de disponibilité peuvent être mappés différemment pour chaque compte.



Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d'une même zone de disponibilité. S3 Express One Zone est conçu pour garantir une disponibilité de 99,95 % dans une seule zone de disponibilité et est soutenu par le contrat de niveau de service Amazon S3. Pour plus d'informations, consultez Zone de disponibilité unique.

S3 Express One Zone est pris en charge dans les régions et zones de disponibilité suivantes :

Régions et zones de disponibilité prises en charge par S3 Express One Zone

Nom de la région	Code région	ID de zone de disponibilité
USA Est (Virginie du Nord)	`us-east-1`	`use1-az4`
		`use1-az5`
		`use1-az6`
USA Ouest (Oregon)	`us-west-2`	`usw2-az1`
		`usw2-az3`
		`usw2-az4`

Nom de la région	Code région	ID de zone de disponibilité
Asie Pacifique (Tokyo)	`ap-northeast-1`	`apne1-az1`
		`apne1-az4`
Europe (Stockholm)	`eu-north-1`	`eun1-az1`
		`eun1-az2`
		`eun1-az3`

Points de terminaison régionaux et zonaux

Pour accéder aux points de terminaison régionaux et zonaux Amazon S3 Express One Zone depuis votre cloud privé virtuel (VPC), vous pouvez utiliser des points de terminaison de VPC de passerelle. Après avoir créé un point de terminaison de passerelle, vous pouvez l'ajouter comme cible dans votre table de routage pour le trafic destiné à S3 Express One Zone depuis votre VPC. Il n'y a pas de frais supplémentaires pour l'utilisation de points de terminaison de passerelle. Pour plus d'informations sur la manière de configurer des points de terminaison de VPC de passerelle, consultez Mise en réseau pour S3 Express One Zone.

Lorsque vous utilisez S3 Express One Zone, les opérations d'API de niveau compartiment (plan de contrôle) sont disponibles via un point de terminaison régional et sont appelées opérations d'API de point de terminaison régional. `CreateBucket` et `DeleteBucket` sont des exemples d'opérations d'API de point de terminaison régional.

Après avoir créé un compartiment de répertoire, vous pouvez utiliser Zonal (opérations d'API au niveau de l'objet ou du point de terminaison du plan de données) pour télécharger et gérer les objets de votre compartiment de répertoire. Les opérations d'API de point de terminaison zonal sont disponibles via un point de terminaison zonal. `PutObject` et `CopyObject` sont des exemples d'opérations d'API zonales.

Opérations d'API S3 Express One Zone

La classe de stockage Amazon S3 Express One Zone prend en charge les opérations d'API de point de terminaison régional (niveau compartiment ou plan de contrôle) et zonal (niveau objet ou plan de

données). Pour plus d'informations, consultez Mise en réseau pour S3 Express One Zone et Points de terminaison et points de terminaison de VPC de passerelle.

Opérations d'API de point de terminaison régional

Les opérations d'API de point de terminaison régional suivantes sont prises en charge pour S3 Express One Zone :

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Opérations d'API de point de terminaison zonal

Les opérations d'API de point de terminaison zonal suivantes sont prises en charge pour S3 Express One Zone :

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads

- ListParts
- UploadPart
- UploadPartCopy

Gestion de l'accès aux données avec les points d'accès Amazon S3

Les points d'accès Amazon S3 simplifient l'accès aux données pour tout AWS service ou application client qui stocke des données dans S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet S3, comme par exemple `GetObject` et `PutObject`. Chaque point d'accès dispose d'autorisations et de contrôles réseau distincts que S3 applique pour toute demande effectuée via ce point d'accès. Chaque point d'accès applique une stratégie de point d'accès personnalisée qui fonctionne conjointement avec la stratégie de compartiment associée au compartiment sous-jacent. Vous pouvez configurer n'importe quel point d'accès pour accepter uniquement les demandes provenant d'un cloud privé virtuel (VPC) afin de restreindre l'accès aux données Amazon S3 à un réseau privé. Vous pouvez également configurer des paramètres de blocage de l'accès public personnalisés pour chaque point d'accès.

> **Note**
>
> - Vous ne pouvez utiliser que des points d'accès pour effectuer des opérations sur des objets. Vous ne pouvez pas utiliser les points d'accès pour effectuer d'autres opérations Amazon S3, telles que la modification ou la suppression de compartiments. Pour obtenir la liste complète des opérations S3 qui prennent en charge les points d'accès, veuillez consulter Compatibilité des points d'accès avec les AWS services.
> - Les points d'accès fonctionnent avec certains AWS services et fonctionnalités, mais pas tous. Par exemple, vous ne pouvez pas configurer la réplication entre régions pour qu'elle fonctionne via un point d'accès. Pour obtenir la liste complète des AWS services compatibles avec les points d'accès S3, consultezCompatibilité des points d'accès avec les AWS services.

Cette section explique comment utiliser les points d'accès Amazon S3. Pour plus d’informations sur l’utilisation des compartiments, consultez Présentation des compartiments. Pour en savoir plus sur l'utilisation des objets, consultez Présentation des objets Amazon S3.

Rubriques

- Création de points d'accès
- Utilisation des points d'accès
- Limites et restrictions des points d'accès

Configuration des stratégies IAM pour l'utilisation des points d'accès

Les points d'accès Amazon S3 prennent en charge les politiques de ressources AWS Identity and Access Management (IAM) qui vous permettent de contrôler l'utilisation du point d'accès par ressource, par utilisateur ou selon d'autres conditions. Pour qu'une application ou un utilisateur puisse accéder à des objets via un point d'accès, il faut que le point d'accès et le compartiment sous-jacent autorisent la demande.

> **Important**
>
> L'ajout d'un point d'accès S3 à un compartiment ne modifie pas le comportement du compartiment lorsqu'on y accède directement par le nom du compartiment ou le nom Amazon Resource Name (ARN). Toutes les opérations existantes sur le compartiment continueront de fonctionner comme auparavant. Les restrictions que vous incluez dans une stratégie de point d'accès s'appliquent uniquement aux demandes effectuées via ce point d'accès.

Lorsque vous utilisez des politiques de ressources IAM, veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions AWS Identity and Access Management Access Analyzer avant d'enregistrer votre politique. IAM Access Analyzer exécute des vérifications de politiques pour valider votre politique par rapport à la grammaire de politique et aux bonnes pratiques IAM. Ces vérifications génèrent des résultats et fournissent des recommandations pour vous aider à créer des stratégies fonctionnelles et conformes aux bonnes pratiques en matière de sécurité.

Pour en savoir plus sur la validation des politiques à l'aide d'IAM Access Analyzer, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM. Pour afficher la liste des avertissements, erreurs et suggestions renvoyés par IAM Access Analyzer, consultez la Référence de vérification de stratégie IAM Access Analyzer.

Exemples de stratégie de point d'accès

Les exemples suivants montrent comment créer des stratégies IAM pour contrôler les demandes effectuées via un point d'accès.

> **Note**
>
> Les autorisations accordées dans une politique de point d'accès ne sont effectives que si le compartiment sous-jacent autorise également le même accès. Vous pouvez y parvenir de deux façons :
>
> 1. (Recommandé) Déléguer le contrôle d'accès du compartiment au point d'accès, comme décrit à la section Délégation du contrôle d'accès aux points d'accès.
> 2. Ajoutez les mêmes autorisations contenues dans la stratégie de point d'accès à la stratégie du compartiment sous-jacent. Le premier exemple de politique de point d'accès montre comment modifier la politique de compartiment sous-jacente pour autoriser l'accès nécessaire.

Example 1 : octroi de politique de point d'accès

La politique de point d'accès suivante accorde à l'utilisateur IAM *Jane* dans le compte *123456789012* des autorisations GET et PUT pour des objets avec le préfixe *Jane/* via le point d'accès *my-access-point* dans le compte *123456789012*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Jane"
        },
        "Action": ["s3:GetObject", "s3:PutObject"],
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/Jane/*"
    }]
}
```

> **Note**
>
> Pour que la stratégie de point d'accès accorde effectivement l'accès à *Jane*, il faut que le compartiment sous-jacent autorise également le même accès à *Jane*. Vous pouvez déléguer le contrôle d'accès du compartiment au point d'accès comme décrit à la section Délégation du contrôle d'accès aux points d'accès. Vous pouvez également ajouter la stratégie suivante au compartiment sous-jacent pour accorder les autorisations nécessaires à Jane. Notez que l'entrée `Resource` diffère entre les stratégies de point d'accès et de compartiment.
>
> ```
> {
>     "Version": "2012-10-17",
>     "Statement": [
>     {
>         "Effect": "Allow",
>         "Principal": {
>             "AWS": "arn:aws:iam::123456789012:user/Jane"
>         },
>         "Action": ["s3:GetObject", "s3:PutObject"],
>         "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/Jane/*"
>     }]
> }
> ```

Example 2 : politique de point d'accès avec condition d'étiquette

La politique de point d'accès suivante accorde à l'utilisateur IAM *Mateo* dans le compte *123456789012* des autorisations pour des objets GET via le point d'accès *my-access-point* du compte *123456789012* dont la valeur de la clé d'étiquette *data* est définie sur *finance*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect":"Allow",
        "Principal" : {
            "AWS": "arn:aws:iam::123456789012:user/Mateo"
        },
        "Action":"s3:GetObject",
```

```
        "Resource" : "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/*",
        "Condition" : {
            "StringEquals": {
                "s3:ExistingObjectTag/data": "finance"
            }
        }
    }]
}
```

Example 3 : politique de point d'accès permettant d'établir la liste des compartiments

La politique de point d'accès suivante permet à l'utilisateur IAM `Arnav` dans le compte *123456789012* d'afficher les objets contenus dans le point d'accès sous-jacent du compartiment *my-access-point* dans le compte *123456789012*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Arnav"
        },
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"
    }]
}
```

Example 4 :politique de contrôle des services

La stratégie de contrôle de service suivante exige que tous les nouveaux points d'accès soient créés avec une origine de réseau de cloud privé virtuel (VPC). Lorsque cette stratégie est mise en place, les utilisateurs de votre organisation ne peuvent pas créer de nouveaux points d'accès accessibles à partir d'Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Deny",
```

```
        "Action": "s3:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    }]
}
```

Example 5 : stratégie de compartiment pour limiter les opérations S3 aux origines réseau VPC

La stratégie de compartiment suivante limite l'accès à toutes les opérations d'objet S3 pour le compartiment *example-s3-bucket* aux points d'accès ayant une origine réseau VPC.

> **Important**
>
> Avant d'utiliser une instruction comme celle présentée dans cet exemple, assurez-vous que vous n'avez pas besoin d'utiliser des fonctions qui ne sont pas prises en charge par les points d'accès, comme la réplication entre régions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:BypassGovernanceRetention",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersion",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
```

```
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionAcl",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject"
            ],
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:AccessPointNetworkOrigin": "VPC"
                }
            }
        }
    ]
}
```

Clés de condition

Les points d'accès S3 comportent des clés de condition que vous pouvez utiliser dans les politiques IAM pour contrôler l'accès à vos ressources. Les clés de condition suivantes ne représentent qu'une partie d'une politique IAM. Pour obtenir des exemples complets de politiques, consultez Exemples de stratégie de point d'accès, the section called “Délégation du contrôle d'accès aux points d'accès” et the section called “Octroi d'autorisations pour les points d'accès intercompte”.

`s3:DataAccessPointArn`

Cet exemple présente une chaîne que vous pouvez utiliser pour établir une correspondance sur un ARN de point d'accès. L'exemple suivant correspond à tous les points d'accès Compte AWS *123456789012*de la région *us-west-2*:

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
}
```

`s3:DataAccessPointAccount`

Cet exemple présente un opérateur de chaîne que vous pouvez utiliser pour établir une correspondance sur l'ID de compte du propriétaire d'un point d'accès. L'exemple suivant correspond à tous les points d'accès appartenant au Compte AWS *123456789012*.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
}
```

`s3:AccessPointNetworkOrigin`

Cet exemple présente un opérateur de chaîne que vous pouvez utiliser pour faire correspondre sur l'origine du réseau, soit `Internet`, soit `VPC`. L'exemple suivant ne correspond qu'aux points d'accès ayant une origine VPC.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Pour plus d'informations sur l'utilisation des clés de condition avec Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Délégation du contrôle d'accès aux points d'accès

Vous pouvez déléguer le contrôle d'accès à un compartiment aux points d'accès du compartiment. Dans l'exemple suivant, la politique de compartiment permet un accès complet à tous les points d'accès appartenant au compte du propriétaire du compartiment. Ainsi, tous les accès à ce compartiment sont contrôlés par les stratégies attachées à ses points d'accès. Nous vous recommandons de configurer vos compartiments de cette façon pour tous les cas d'utilisation qui ne demandent pas d'accès direct au compartiment.

Example 6 : politique de compartiment qui délègue le contrôle d'accès aux points d'accès

```
{
```

```
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Octroi d'autorisations pour les points d'accès intercompte

Pour créer un point d'accès à un compartiment qui appartient à un autre compte, vous devez d'abord créer le point d'accès en spécifiant le nom du compartiment et l'ID du propriétaire du compte. Ensuite, le propriétaire du compartiment doit mettre à jour la politique de compartiment pour autoriser les requêtes du point d'accès. La création d'un point d'accès est similaire à la création d'un DNS CNAME dans la mesure où le point d'accès ne donne pas accès au contenu du compartiment. Tous les accès aux compartiments sont contrôlés par la politique des compartiments. L'exemple de politique de compartiment suivant autorise les requêtes `GET` et `LIST` sur le compartiment depuis un point d'accès appartenant à un Compte AWS de confiance.

Remplacez l'*ARN du bucket* par l'ARN du bucket.

Example 7 — Politique de compartiment déléguant des autorisations à un autre Compte AWS

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : ["s3:GetObject","s3:ListBucket"],
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Access point owner's
 account ID" }
        }
    }]
```

```
}
```

Création de points d'accès

Amazon S3 fournit des fonctionnalités pour créer et gérer des points d'accès. Vous pouvez créer des points d'accès S3 à l'aide des AWS SDK AWS Management Console, AWS Command Line Interface (AWS CLI) ou de l'API REST Amazon S3.

Par défaut, vous pouvez créer jusqu'à 10 000 points d'accès par région pour chacun de vos Comptes AWS. Si vous avez besoin de plus de 10 000 points d'accès pour un seul compte dans une même région, vous pouvez demander une augmentation des quotas de service. Pour plus d'informations sur les quotas de service et la demande d'une augmentation, consultez AWS Service Quotas dans Références générales AWS.

> **Note**
>
> Dans la mesure où vous pouvez publier le nom de votre point d'accès pour que d'autres utilisateurs puissent utiliser le point d'accès, évitez d'inclure des informations sensibles dans le nom du point d'accès. Les noms des points d'accès sont publiés dans une base de données accessible au public, appelée Domain Name System (DNS).

Règles relatives à l'attribution de noms pour les points d'accès Amazon S3

Les noms de point d'accès doivent remplir les conditions suivantes :

- Doit être unique au sein d'une même Compte AWS région
- Ils doivent se conformer aux restrictions d'attribution de noms DNS.
- Ils doivent commencer par un chiffre ou une lettre minuscule.
- Ils doivent comporter entre 3 et 50 caractères.
- Ils ne peuvent pas commencer ou se terminer par un trait d'union (`-`)
- Ils ne peuvent contenir ni traits de soulignement (_), ni lettres majuscules, ni points (`.`)
- Ne peut pas se terminer par le suffixe `-s3alias`. Ce suffixe est réservé aux noms d'alias de point d'accès. Pour plus d’informations, consultez Utilisation d'un alias de type compartiment pour votre point d'accès de compartiment S3.

Pour créer un point d'accès, consultez les rubriques suivantes.

Rubriques

Création d'un point d'accès

Un point d'accès est associé à un seul compartiment Amazon S3. Si vous souhaitez utiliser un bucket dans votre Compte AWS, vous devez d'abord en créer un. Pour en savoir plus sur la création des compartiments, consultez Création, configuration et utilisation des compartiments Amazon S3.

Vous pouvez également créer un point d'accès intercompte associé à un compartiment dans un autre compte Compte AWS, à condition de connaître le nom du compartiment et l'ID du compte du propriétaire du compartiment. Cependant, la création de points d'accès intercompte ne vous donne pas accès aux données du compartiment tant que vous n'avez pas obtenu les autorisations du propriétaire du compartiment. Le propriétaire du compartiment doit accorder au compte du propriétaire du point d'accès (votre compte) l'accès au compartiment par le biais de la stratégie du compartiment. Pour plus d’informations, consultez Octroi d'autorisations pour les points d'accès intercompte.

Par défaut, vous pouvez créer jusqu'à 10 000 points d'accès par région pour chacun de vos Comptes AWS. Si vous avez besoin de plus de 10 000 points d'accès pour un seul compte dans une même région, vous pouvez demander une augmentation des quotas de service. Pour plus d'informations sur les quotas de service et la demande d'une augmentation, consultez AWS Service Quotas dans Références générales AWS.

Les exemples suivants montrent comment créer un point d'accès avec la console AWS CLI et la console S3. Pour plus d'informations sur la création d'un point d'accès à l'aide de l'API REST, consultez CreateAccessPoint dans la référence des API Amazon Simple Storage Service.

Utilisation de la console S3

Pour créer un point d'accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer un point d'accès.
3. Dans le panneau de navigation, choisissez Points d'accès.
4. Dans la page Points d'accès, choisissez Créer un point d'accès.
5. Dans le champ Nom du point d'accès, saisissez le nom que vous souhaitez pour le point d'accès. Pour en savoir plus sur l'attribution de noms aux points d'accès, consultez Règles relatives à l'attribution de noms pour les points d'accès Amazon S3.
6. Dans Nom du compartiment, indiquez le compartiment S3 que vous souhaitez utiliser avec le point d'accès.

   Pour utiliser un compartiment dans votre compte, sélectionnez Choisir un compartiment dans ce compte et saisissez ou recherchez le nom du compartiment.

   Pour utiliser un compartiment dans un autre compte Compte AWS, choisissez Spécifier un compartiment dans un autre compte, puis entrez l' Compte AWS ID et le nom du compartiment.

   > **Note**
   >
   > Si vous utilisez un bucket dans un autre endroit Compte AWS, le propriétaire du bucket doit mettre à jour la politique du bucket afin d'autoriser les demandes provenant du point d'accès. Pour un exemple de stratégie de compartiment, consultez Octroi d'autorisations pour les points d'accès intercompte.

7. Choisissez une Origine du réseau. Si vous choisissez Virtual Private Cloud (VPC), entrez le paramètre VPC ID (ID VPC) que vous souhaitez utiliser avec le point d'accès.

   Pour en savoir plus sur les origines réseau des points d'accès, veuillez consulter Création de points d'accès restreints à un virtual private cloud.
8. Sous Block Public Access settings for this Access Point (Blocage de l'accès public pour ce point d'accès), sélectionnez les paramètres de blocage de l'accès public que vous voulez appliquer au point d'accès. Tous les paramètres de blocage d'accès public sont activés par défaut pour les nouveaux points d'accès. Nous vous recommandons de garder tous les paramètres activés, sauf si vous savez que vous avez un besoin spécifique de désactiver l'un d'entre eux.

> **Note**
>
> Après avoir créé un point d'accès, vous ne pouvez pas modifier ses paramètres de blocage de l'accès public.

Pour en savoir plus sur l'utilisation du blocage de l'accès public d'Amazon S3 avec des points d'accès, veuillez consulter Gestion de l'accès public aux points d'accès.

9. (Facultatif) Sous Access point policy (Stratégie de point d'accès) - facultatif, spécifiez la stratégie de point d'accès. Avant d'enregistrer votre politique, veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions. Pour en savoir plus sur la spécification d'une stratégie de point d'accès, veuillez consulter Exemples de stratégie de point d'accès.
10. Choisissez Create access point (Créer un point d'accès).

À l'aide du AWS CLI

L'exemple de commande suivant crée un point d'accès nommé *example-ap* pour le compartiment *DOC-EXAMPLE-BUCKET* dans le compte *111122223333*. Pour créer le point d'accès, vous envoyez une requête à Amazon S3 qui spécifie les éléments suivants :

- Nom du point d'accès. Pour plus d'informations sur les règles d'attribution de noms, consultez the section called “Règles relatives à l'attribution de noms pour les points d'accès Amazon S3”.
- Le nom du compartiment auquel vous voulez associer le point d'accès.
- L'ID de compte du Compte AWS propriétaire du bucket.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket DOC-EXAMPLE-BUCKET
```

Lorsque vous créez un point d'accès en utilisant un bucket dans un autre Compte AWS, incluez le `--bucket-account-id` paramètre. L'exemple de commande suivant crée un point d'accès dans le Compte AWS *111122223333*, en utilisant le compartiment *DOC-EXAMPLE-BUCKET2*, qui se trouve dans le Compte AWS *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket DOC-EXAMPLE-BUCKET --bucket-account-id 444455556666
```

Création de points d'accès restreints à un virtual private cloud

Lorsque vous créez un point d'accès, vous pouvez choisir de rendre le point d'accès accessible à partir d'Internet, ou vous pouvez spécifier que toutes les demandes effectuées via ce point d'accès doivent provenir d'un virtual private cloud (VPC) spécifique. Un point d'accès accessible depuis Internet est dit avoir une origine réseau d'`Internet`. Il peut être utilisé à partir de n'importe quel endroit sur Internet, sous réserve de toute autre restriction d'accès qui s'appliquent au point d'accès, au compartiment sous-jacent et aux ressources associées, telles que les objets demandés. Un point d'accès accessible uniquement à partir d'un VPC spécifié a une origine réseau de `VPC`, et Amazon S3 rejette toute demande adressée au point d'accès qui ne provient pas de ce VPC.

> **Important**
>
> Vous ne pouvez spécifier l'origine réseau d'un point d'accès qu'au moment de la création du point d'accès. Après avoir créé le point d'accès, vous ne pouvez pas modifier son origine réseau.

Pour limiter un point d'accès à un accès VPC uniquement, vous devez inclure le paramètre `VpcConfiguration` à la demande de création du point d'accès. Dans le paramètre `VpcConfiguration`, vous spécifiez l'ID de VPC que vous souhaitez pouvoir utiliser avec le point d'accès. Si une demande est effectuée via le point d'accès, elle doit provenir du VPC, car Amazon S3 la rejettera dans le cas contraire.

Vous pouvez récupérer l'origine réseau d'un point d'accès à l'aide AWS CLI AWS des SDK ou des API REST. Si un point d'accès a une configuration VPC spécifiée, son origine réseau est `VPC`. Sinon, l'origine réseau du point d'accès est `Internet`.

Example

Exemple : créer un point d'accès restreint à l'accès VPC

L'exemple suivant crée un point d'accès nommé `example-vpc-ap` pour le compartiment `example-bucket` dans le compte `123456789012` qui autorise l'accès uniquement à partir du VPC `vpc-1a2b3c`. L'exemple vérifie ensuite que le nouveau point d'accès a une origine réseau de `VPC`.

AWS CLI

```
aws s3control create-access-point --name example-vpc-ap --account-id 123456789012 --
bucket example-bucket --vpc-configuration VpcId=vpc-1a2b3c
```

```
aws s3control get-access-point --name example-vpc-ap --account-id 123456789012

{
    "Name": "example-vpc-ap",
    "Bucket": "example-bucket",
    "NetworkOrigin": "VPC",
    "VpcConfiguration": {
        "VpcId": "vpc-1a2b3c"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Pour utiliser un point d'accès avec un VPC, vous devez modifier la stratégie d'accès pour votre point de terminaison d'un VPC. Les points de terminaison d'un VPC autorisent la circulation du trafic de votre VPC vers Amazon S3. Ils disposent de politiques de contrôle d'accès qui contrôlent la façon dont les ressources du VPC sont autorisées à interagir avec Amazon S3. Les demandes de votre VPC vers Amazon S3 ne réussissent via un point d'accès que si la politique de point de terminaison d'un VPC accorde l'accès au point d'accès et au compartiment sous-jacent.

> **Note**
>
> Pour rendre les ressources accessibles uniquement au sein d'un VPC, veillez à créer une zone hébergée privée pour votre point de terminaison de VPC. Pour utiliser une zone hébergée privée, modifiez les paramètres de votre VPC de sorte que les attributs de réseau VPC `enableDnsHostnames` et `enableDnsSupport` soient définis sur `true`.

L'exemple de déclaration de stratégie suivant configure un point de terminaison d'un VPC pour autoriser les appels à `GetObject` pour un compartiment nommé `awsexamplebucket1` et un point d'accès nommé `example-vpc-ap`.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*",
            "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
        ]
    }]
}
```

> **Note**
>
> La déclaration `"Resource"` de cet exemple utilise un Amazon Resource Name (ARN) pour spécifier le point d'accès. Pour plus d'informations sur les ARN des points d'accès, consultez Utilisation des points d'accès.

Pour plus d'informations sur les stratégies de point de terminaison d'un VPC, consultez Stratégies de point de terminaison pour Amazon S3 dans le Guide de l'utilisateur Amazon Virtual Private Cloud.

Gestion de l'accès public aux points d'accès

Les points d'accès Amazon S3 prennent en charge des paramètres de blocage de l'accès public indépendants pour chaque point d'accès. Lorsque vous créez un point d'accès, vous pouvez spécifier les paramètres de blocage d'accès public qui s'appliquent à ce point d'accès. Pour toute demande effectuée via un point d'accès, Amazon S3 évalue les paramètres de blocage d'accès public pour ce point d'accès, le compartiment sous-jacent et le compte du propriétaire du compartiment. Si l'un de ces paramètres indique que la demande doit être bloquée, Amazon S3 rejette la demande.

Pour plus d'informations sur le blocage de l'accès public S3, consultez Blocage de l'accès public à votre stockage Amazon S3.

> **Important**
>
> - Tous les paramètres de blocage d'accès public sont activés par défaut pour les points d'accès. Vous devez désactiver explicitement tous les paramètres que vous ne souhaitez pas appliquer à un point d'accès.
> - Actuellement, Amazon S3 ne prend pas en charge la modification des paramètres de blocage de l'accès public d'un point d'accès après que ce point d'accès a été créé.

Example

Exemple : Créer un point d'accès avec des paramètres de blocage d'accès public personnalisés

Cet exemple montre comment créer un point d'accès nommé `example-ap` pour le compartiment `example-bucket` dans le compte `123456789012` avec des paramètres de blocage d'accès public non définis par défaut. L'exemple extrait ensuite la configuration du nouveau point d'accès pour vérifier ses paramètres de blocage d'accès public.

AWS CLI

```
aws s3control create-access-point --name example-ap --account-id
 123456789012 --bucket example-bucket --public-access-block-configuration
 BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

```
aws s3control get-access-point --name example-ap --account-id 123456789012

{
    "Name": "example-ap",
    "Bucket": "example-bucket",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
```

```
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Utilisation des points d'accès

Vous pouvez accéder aux objets d'un compartiment Amazon S3 via un point d'accès à l' AWS Management Console aide AWS CLI AWS des SDK ou des API REST S3.

Les points d'accès possèdent des noms Amazon Resource Name (ARN). Les noms Amazon Resource Name (ARN) de point d'accès sont semblables à ceux de compartiment, mais ils sont saisis explicitement et encodent la Région du point d'accès et l'ID de Compte AWS du propriétaire du point d'accès. Pour plus d'informations sur l'utilisation des ARN, consultez Amazon Resource Name (ARN) dans le Références générales AWS.

Les ARN des points d'accès utilisent le format `arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`resource`*. Exemples :

- arn:aws:s3:us-west-2:123456789012:accesspoint/test représente le point d'accès nommé `test`, appartenant au compte `123456789012` dans la Région `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/* représente tous les points d'accès sous le compte `123456789012` dans la Région `us-west-2`.

Les ARN des objets accessibles via un point d'accès utilisent le format `arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`access-point-name`*`/object/`*`resource`*. Exemples :

- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01 représente l'objet `unit-01`, accessible via le point d'accès nommé `test`, appartenant au compte `123456789012` dans la Région `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/* représente tous les objets pour le point d'accès `test`, dans le compte `123456789012` dans la Région `us-west-2`.
- arn:aws:s3:us-west-2:123456789012:accesspoint/test/object/unit-01/finance/* représente tous les objets sous le préfixe `unit-01/finance/` pour point d'accès `test`, dans le compte `123456789012` dans la Région `us-west-2`.

Accès à un compartiment via les points d'accès S3

Les points d'accès S3 prennent uniquement en charge l' virtual-host-style adressage. Pour faire référence à un compartiment via un point d’accès, utilisez le format suivant.

```
https://AccessPointName-AccountId.s3-accesspoint.region.amazonaws.com
```

> **Note**
>
> - Si le nom de votre point d'accès inclut des tirets (-), incluez les tirets dans l'URL et insérez un autre tiret avant l'ID du compte. Par exemple, pour utiliser un point d'accès nommé `finance-docs` appartenant au compte `123456789012` dans la région `us-west-2`, l'URL appropriée serait `https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
> - Les points d'accès S3 ne prennent pas en charge l'accès par HTTP, uniquement l'accès sécurisé par HTTPS.

Rubriques

- Surveillance et journalisation des points d'accès
- Utilisation des points d'accès Amazon S3 dans la console Amazon S3
- Utilisation d'un alias de type compartiment pour votre point d'accès de compartiment S3
- Utilisation de points d'accès avec les opérations compatibles avec Amazon S3

Si vous disposez d'un cloud privé virtuel (VPC), consultez Gestion de l'accès à Amazon S3 avec des points de terminaison de VPC et des points d'accès S3.

Surveillance et journalisation des points d'accès

Amazon S3 enregistre les demandes effectuées via les points d'accès et les demandes adressées aux API qui gèrent les points d'accès, telles que `CreateAccessPoint` et `GetAccessPointPolicy`. Pour surveiller et gérer les modèles d'utilisation, vous pouvez également configurer les métriques de demande Amazon CloudWatch Logs pour les points d'accès.

Rubriques

- CloudWatch métriques de demande
- Journaux des demandes

CloudWatch métriques de demande

Pour comprendre et améliorer les performances des applications qui utilisent des points d'accès, vous pouvez utiliser CloudWatch les métriques de demande Amazon S3. Les métriques de demandes vous aident à contrôler les demandes Amazon S3 pour identifier rapidement les problèmes opérationnels et agir en conséquence.

Par défaut, ces métriques de demandes sont disponibles au niveau du compartiment . Toutefois, vous pouvez définir un filtre pour les métriques de demandee à l'aide d'un préfixe partagé, de balises d'objet ou d'un point d'accès. Lorsque vous créez un filtre de point d'accès, la configuration des métriques de demandes inclut les demandes vers le point d'accès que vous spécifiez. Ce point d'accès vous permet de recevoir les métriques, de définir les alarmes et d'accéder aux tableaux de bord pour afficher les opérations effectuées en temps réel.

Vous devez activer les métriques de demandes en les configurant dans la console ou en utilisant l'API Amazon S3. Ces métriques sont disponibles à des intervalles d'une minute après une latence pour le traitement. Les métriques de demande sont facturées au même tarif que les métriques CloudWatch personnalisées. Pour plus d'informations, consultez les CloudWatch tarifs Amazon.

Pour créer une configuration de métriques de demandes qui filtre par point d'accès, consultez Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès.

Journaux des demandes

Vous pouvez enregistrer les demandes effectuées via les points d'accès et les demandes adressées aux API qui gèrent les points d'accès, telles que `CreateAccessPoint` et `GetAccessPointPolicy`, en utilisant le journal des accès au serveur et AWS CloudTrail.

CloudTrail les entrées du journal pour les demandes effectuées via les points d'accès incluent l'ARN du point d'accès dans la `resources` section du journal.

Par exemple, supposons que vous ayez la configuration suivante :

- Un compartiment nommé `DOC-EXAMPLE-BUCKET1` dans la Région `us-west-2` qui contient un objet `my-image.jpg`

- Un point d'accès nommé `my-bucket-ap` associé à `DOC-EXAMPLE-BUCKET1`
- Un Compte AWS identifiant de `123456789012`

L'exemple suivant montre la `resources` section d'une entrée de CloudTrail journal pour la configuration précédente :

```
"resources": [
        {"type": "AWS::S3::Object",
            "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/my-image.jpg"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
        }
    ]
```

Pour plus d'informations sur les journaux d'accès au serveur S3, consultez Enregistrement de demandes avec journalisation des accès au serveur. Pour plus d'informations AWS CloudTrail, voir Qu'est-ce que c'est AWS CloudTrail ? dans le guide de AWS CloudTrail l'utilisateur.

Utilisation des points d'accès Amazon S3 dans la console Amazon S3

Cette section explique comment gérer et utiliser vos points d'accès Amazon S3 à l'aide de la AWS Management Console. Avant de commencer, accédez à la page des détails du point d'accès que vous souhaitez gérer ou utiliser, comme décrit dans la procédure suivante.

Rubriques

- Répertorier les points d'accès pour votre compte
- Répertorier les points d'accès pour un compartiment
- Affichage des détails de configuration d'un point d'accès
- Utilisation d'un point d'accès
- Affichage des paramètres d'un point d'accès pour le blocage de l'accès public
- Modification d'une stratégie de point d'accès

- Suppression d'un point d'accès

Répertorier les points d'accès pour votre compte

Pour répertorier tous les points d'accès créés dans votre Compte AWS

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région pour laquelle vous souhaitez répertorier les points d'accès.
3. Dans le panneau de navigation situé sur le côté gauche de la console, choisissez Access points (Points d'accès).
4. Sur la page des points d'accès, sous Points d'accès, consultez les points d'accès de votre Région AWS.
5. (Facultatif) Recherchez les points d'accès par nom en saisissant un nom dans le champ de texte situé à côté du menu déroulant Region (Région).
6. Choisissez le nom du point d'accès que vous souhaitez gérer ou utiliser.

Répertorier les points d'accès pour un compartiment

Pour répertorier tous les points d'accès Compte AWS présents dans un seul compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom de la région actuellement affichée, Région AWS puis choisissez la région pour laquelle vous souhaitez répertorier les points d'accès.
3. Dans le panneau de navigation sur le côté gauche de la console, choisissez Buckets (Compartiments).
4. Dans la page Buckets (Compartiments) sélectionnez le nom du compartiment dont vous souhaitez répertorier les points d'accès.
5. Sur la page des détails du compartiment, choisissez l'onglet Access points (Points d'accès).
6. Choisissez le nom du point d'accès que vous souhaitez gérer ou utiliser.

Affichage des détails de configuration d'un point d'accès

1. Accédez à la page des détails du point d'accès pour le point d'accès dont vous souhaitez afficher les détails, comme décrit à la section Répertorier les points d'accès pour votre compte.
2. Sous Access point overview (Présentation du point d'accès), affichez les détails de configuration et les propriétés du point d'accès sélectionné.

Utilisation d'un point d'accès

1. Accédez à la page des détails du point d'accès pour le point d'accès que vous souhaitez utiliser, comme décrit à la section Répertorier les points d'accès pour votre compte.
2. Sous l'onglet Objects (Objets) choisissez le nom d'un ou des objets auxquels vous souhaitez accéder via le point d'accès. Sur les pages des opérations d'objet, la console affiche une étiquette au-dessus du nom de votre compartiment qui indique le point d'accès que vous utilisez actuellement. Lorsque vous utilisez le point d'accès, vous ne pouvez effectuer que les opérations d'objet permises par les autorisations du point d'accès.

> **Note**
>
> - La vue de la console affiche toujours tous les objets du compartiment. L'utilisation d'un point d'accès comme cette procédure le décrit limite les opérations que vous pouvez effectuer sur ces objets, mais pas si vous pouvez voir qu'ils existent dans le compartiment.
> - La console de gestion S3 ne prend pas en charge l'utilisation de points d'accès Virtual Private Cloud (VPC) pour accéder aux ressources de compartiment. Pour accéder aux ressources du bucket depuis un point d'accès VPC, utilisez les AWS SDK AWS CLI ou les API REST Amazon S3.

Affichage des paramètres d'un point d'accès pour le blocage de l'accès public

1. Accédez à la page des détails du point d'accès dont vous souhaitez afficher les paramètres, comme décrit à la section Répertorier les points d'accès pour votre compte.
2. Choisissez Permissions.
3. Sous Access point policy (Stratégie de point d'accès), vérifiez les paramètres de blocage de l'accès public du point d'accès.

> **Note**
>
> Vous ne pouvez pas modifier les paramètres de blocage de l'accès public pour un point d'accès après sa création.

Modification d'une stratégie de point d'accès

1. Accédez à la page des détails du point d'accès dont vous souhaitez modifier la stratégie, comme décrit à la section Répertorier les points d'accès pour votre compte.
2. Choisissez Permissions.
3. Sous Access point policy (Stratégie de point d'accès), choisissez Edit (Modifier).
4. Entrez la stratégie de point d'accès dans le champ de texte. La console affiche automatiquement le nom Amazon Resource Name (ARN) du point d'accès, que vous pouvez utiliser dans la stratégie.

Suppression d'un point d'accès

1. Accédez à la liste des points d'accès pour votre compte ou pour un compartiment spécifique, comme décrit à la section Répertorier les points d'accès pour votre compte.
2. Sélectionnez le bouton d'option en regard du nom du point d'accès à supprimer.
3. Sélectionnez Delete.
4. Confirmez que vous souhaitez supprimer votre point d'accès en entrant son nom dans le champ de texte qui s'affiche, puis choisissez Delete (Supprimer).

Utilisation d'un alias de type compartiment pour votre point d'accès de compartiment S3

Lorsque vous créez un point d'accès, Amazon S3 génère automatiquement un alias que vous pouvez utiliser pour l'accès aux données au lieu d'un nom de compartiment Amazon S3. Vous pouvez utiliser cet alias de point d'accès à la place d'un Amazon Resource Name (ARN) pour les opérations de plan de données de point d'accès. Pour obtenir la liste de ces opérations, veuillez consulter la page Compatibilité des points d'accès avec les AWS services.

Voici un exemple d'ARN et d'alias de point d'accès pour un point d'accès nommé *my-access-point*.

- ARN – `arn:aws:s3:region:account-id:accesspoint/my-access-point`
- Alias de point d'accès – `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias`

Pour plus d'informations sur l'utilisation des ARN, consultez Amazon Resource Name (ARN) dans le Références générales AWS.

Noms d'alias du point d'accès

Un nom d'alias de point d'accès est créé dans le même espace de noms qu'un compartiment Amazon S3. Ce nom d'alias est généré automatiquement et ne peut pas être modifié. Un nom d'alias de point d'accès répond à toutes les exigences d'un nom de compartiment Amazon S3 valide et comprend les parties suivantes :

access point prefix`-`*metadata*`-s3alias`

> **Note**
>
> Le suffixe `-s3alias` est réservé aux noms d'alias de point d'accès et ne peut pas être utilisé pour les noms de compartiment ou de point d'accès. Pour plus d'informations sur les règles d'attribution de noms des compartiments Amazon S3, consultez Règles de dénomination de compartiment.

Cas d'utilisation et limitations des alias de point d'accès

Lorsque vous adoptez des points d'accès, vous pouvez utiliser des noms d'alias de point d'accès sans nécessiter d'importantes modifications du code.

Lorsque vous créez un point d'accès, Amazon S3 génère automatiquement un nom d'alias de point d'accès, comme illustré dans l'exemple suivant. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control create-access-point --bucket example-s3-bucket1 --name my-access-point --
account-id 111122223333
{
```

```
    "AccessPointArn":
    "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
    "Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}
```

Vous pouvez utiliser ce nom d'alias de point d'accès plutôt qu'un nom de compartiment Amazon S3 pour toutes les opérations de plan de données. Pour obtenir la liste de ces opérations, veuillez consulter la page Compatibilité des points d'accès avec les AWS services.

L' AWS CLI exemple de `get-object` commande suivant utilise l'alias du point d'accès du compartiment pour renvoyer des informations sur l'objet spécifié. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api get-object --bucket my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-
s3alias --key dir/my_data.rtf my_data.rtf

{
    "AcceptRanges": "bytes",
    "LastModified": "2020-01-08T22:16:28+00:00",
    "ContentLength": 910,
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
    "VersionId": "null",
    "ContentType": "text/rtf",
    "Metadata": {}
}
```

Limites

- Les alias ne peuvent pas être configurés par les clients.
- Les alias ne peuvent pas être supprimés, modifiés ni désactivés sur un point d'accès.
- Vous pouvez utiliser ce nom d'alias de point d'accès à la place d'un nom de compartiment Amazon S3 dans certaines opérations de plan de données. Pour obtenir la liste de ces opérations, veuillez consulter la page Compatibilité des points d'accès avec les opérations S3.
- Vous ne pouvez pas utiliser un nom d'alias de point d'accès pour les opérations relatives au plan de contrôle Amazon S3. Pour obtenir la liste des opérations relatives au plan de contrôle Amazon S3, veuillez consulter la section Contrôle Amazon S3 dans la Référence des API Amazon Simple Storage Service.
- Vous ne pouvez pas utiliser les alias de point d'accès S3 comme source ou destination pour les opérations Move dans la console Amazon S3.

- Les alias ne peuvent pas être utilisés dans les AWS Identity and Access Management politiques (IAM).
- Les alias ne peuvent pas être utilisés comme destination de journalisation pour les journaux d'accès au serveur S3.
- Les alias ne peuvent pas être utilisés comme destination de journalisation pour les AWS CloudTrail journaux.
- Amazon SageMaker GroundTruth ne prend pas en charge les alias de point d'accès.

Utilisation de points d'accès avec les opérations compatibles avec Amazon S3

Les exemples suivants montrent comment utiliser des points d'accès avec des opérations compatibles dans Amazon S3.

Rubriques

Compatibilité des points d'accès avec les AWS services

Les alias de points d'accès Amazon S3 permettent aux applications qui nécessitent un nom de compartiment S3 d'utiliser facilement un point d'accès. Vous pouvez utiliser des alias de point d'accès S3 à chaque fois que vous utilisez des noms de compartiment S3 pour accéder aux données dans S3. Pour plus d’informations, consultez Cas d'utilisation et limitations des alias de point d'accès.

Compatibilité des points d'accès avec les opérations S3

Vous pouvez utiliser des points d'accès pour accéder à un compartiment à l'aide du sous-ensemble des API Amazon S3 suivant. Toutes les opérations répertoriées ci-dessous peuvent accepter des ARN ou des alias de point d'accès :

Opérations S3

- AbortMultipartUpload
- CompleteMultipartUpload
- CopyObject (copies dans une même Région uniquement)
- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetBucketAcl
- GetBucketCors
- GetBucketLocation
- GetBucketNotificationConfiguration
- GetBucketPolicy
- GetObject
- GetObjectAcl
- GetObjectAttributes
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- ListObjectVersions
- ListParts
- Presign

- PutObject
- PutObjectLegalHold
- PutObjectRetention
- PutObjectAcl
- PutObjectTagging
- RestoreObject
- UploadPart
- UploadPartCopy (copies dans une même Région uniquement)

Demander un objet via un point d'accès

L'exemple suivant demande l'objet `my-image.jpg` via le point d'accès `prod` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`, et enregistre le fichier téléchargé en tant que `download.jpg`.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-
west-2:123456789012:accesspoint/prod download.jpg
```

Charger un objet par le biais d'un alias de point d'accès

L'exemple suivant télécharge l'objet `my-image.jpg` par le biais de l'alias de point d'accès `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-
s3alias --key my-image.jpg --body my-image.jpg
```

Supprimer un objet via un point d'accès

L'exemple suivant supprime l'objet `my-image.jpg` via le point d'accès `prod` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg
```

Répertorier des objets par le biais d'un alias de point d'accès

L'exemple suivant répertorie les objets par le biais d'un alias de point d'accès `my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-
hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Ajouter un ensemble de balises à un objet via un point d'accès

L'exemple suivant ajoute un ensemble de balises à l'objet existant `my-image.jpg` via le point d'accès `prod` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`.

AWS CLI

```
aws s3api put-object-tagging --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/
prod --key my-image.jpg --tagging TagSet=[{Key="finance",Value="true"}]
```

Octroyer des autorisations d'accès via un point d'accès à l'aide d'une liste ACL

L'exemple suivant applique une liste de contrôle d'accès à un objet existant `my-image.jpg` via le point d'accès `prod` appartenant à l'ID de compte `123456789012` dans la Région `us-west-2`.

AWS CLI

```
aws s3api put-object-acl --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg --acl private
```

Limites et restrictions des points d'accès

Les points d'accès Amazon S3 comportent les limites et restrictions suivantes :

- Chaque point d'accès est associé à un seul compartiment, que vous devez spécifier lors de la création du point d'accès. Une fois que vous avez créé un point d'accès, vous ne pouvez pas l'associer à un autre compartiment. Toutefois, vous pouvez supprimer un point d'accès, puis en créer un autre portant le même nom et associer ce nouveau point d'accès à un compartiment différent.
- Les noms des points d'accès doivent répondre à certaines conditions. Pour en savoir plus sur l'attribution de noms aux points d'accès, consultez Règles relatives à l'attribution de noms pour les points d'accès Amazon S3.
- Après avoir créé un point d'accès, vous ne pouvez pas modifier sa configuration VPC (virtual private cloud).
- Les stratégies de point d'accès sont limitées à une taille de 20 Ko.
- Vous pouvez créer un maximum de 10 000 points d'accès Compte AWS par région. Si vous avez besoin de plus de 10 000 points d'accès pour un seul compte dans une même région, vous pouvez demander une augmentation des quotas de service. Pour plus d'informations sur les quotas de service et la demande d'une augmentation, consultez AWS Service Quotas dans Références générales AWS.
- Régions AWS Lorsque vous avez plus de 1 000 points d'accès, vous ne pouvez pas rechercher un point d'accès par son nom dans la console Amazon S3.
- Vous ne pouvez pas utiliser un point d'accès comme destination pour la réplication S3. Pour plus d'informations sur la réplication, consultez Vue d'ensemble de la réplication d'objets.
- Vous ne pouvez pas utiliser les alias de point d'accès S3 comme source ou destination pour les opérations Move dans la console Amazon S3.
- Vous ne pouvez adresser les points d'accès qu'à l'aide d' virtual-host-style URL. Pour plus d'informations sur l' virtual-host-style adressage, consultezAccès à un compartiment Amazon S3 et liste des compartiments.
- Les opérations d'API qui contrôlent la fonctionnalité des points d'accès (par exemple, `PutAccessPoint` et `GetAccessPointPolicy`) ne prennent pas en charge les appels intercomptes.
- Vous devez utiliser AWS Signature Version 4 lorsque vous envoyez des demandes à un point d'accès à l'aide des API REST. Pour plus d'informations sur l'authentification des demandes,

consultez Authentification des demandes (AWS Signature version 4) dans le manuel Amazon Simple Storage Service API Reference.

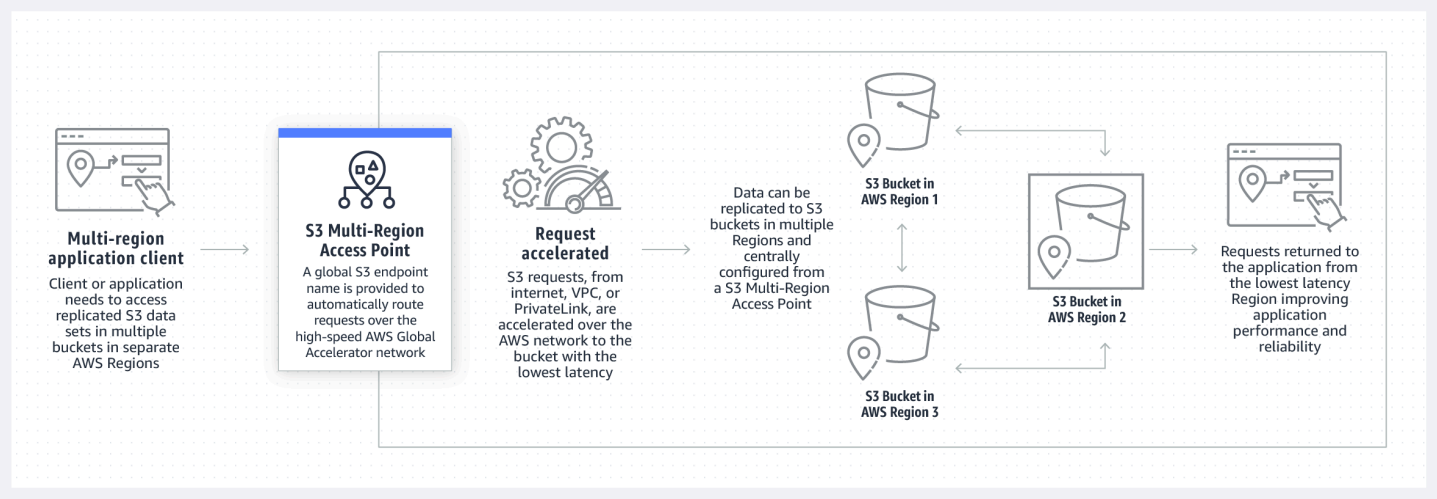
- Les points d'accès ne prennent en charge que les demandes via HTTPS. Amazon S3 répondra automatiquement par une redirection HTTP pour toutes les demandes effectuées via HTTP, afin de passer à HTTPS.
- Les points d'accès ne prennent pas en charge l'accès anonyme.
- Les points d'accès intercompte n'accordent pas l'accès aux données tant que vous n'avez pas obtenu les autorisations du propriétaire du compartiment. Le propriétaire du compartiment conserve toujours le contrôle ultime sur l'accès aux données et doit mettre à jour la politique du compartiment pour autoriser les demandes provenant du point d'accès intercompte. Pour afficher un exemple de politique de compartiment, consultez Configuration des stratégies IAM pour l'utilisation des points d'accès.
- Lorsque vous consultez un point d'accès intercompte dans la console Amazon S3, la colonne Accès affiche Inconnu. La console Amazon S3 ne peut pas déterminer si un accès public est accordé au compartiment et aux objets associés. À moins que vous n'ayez besoin d'une configuration publique pour un cas d'utilisation spécifique, nous vous recommandons, ainsi qu'au propriétaire du compartiment, de bloquer tout accès public au point d'accès et au compartiment. Pour plus d’informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Points d'accès multi-régions dans Amazon S3

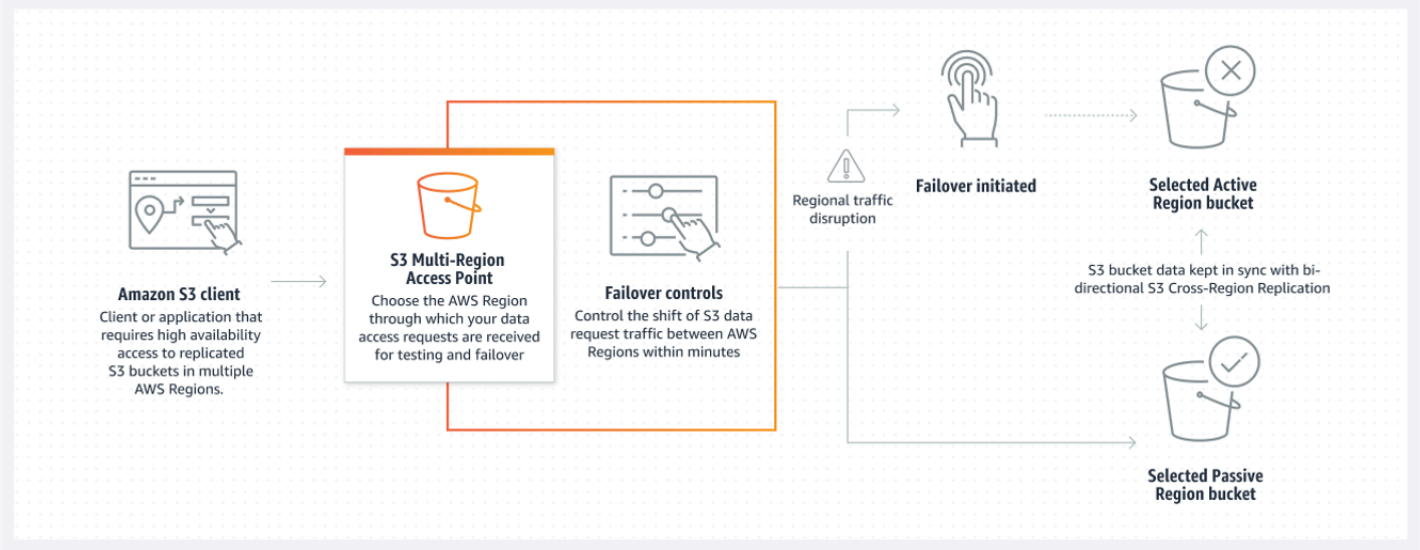
Les points d'accès multi-régions Amazon S3 fournissent un point de terminaison global que les applications peuvent utiliser pour traiter les demandes provenant de compartiments S3 situés dans plusieurs Régions AWS. Vous pouvez utiliser des points d'accès multi-régions pour créer des applications multi-régions avec la même architecture utilisée dans une seule région, puis exécuter ces applications partout dans le monde. Au lieu d'envoyer les demandes sur l'Internet public encombré, les points d'accès multi-régions offrent une résilience de réseau intégrée avec une accélération des demandes Internet vers Amazon S3. Les demandes d'application adressées à un point d'accès global multirégional sont utilisées AWS Global Acceleratorpour acheminer automatiquement sur le réseau AWS mondial le compartiment S3 le plus proche avec un statut de routage actif.

Lorsque vous créez un point d'accès multirégional, vous spécifiez un ensemble d'emplacements Régions AWS dans lesquels vous souhaitez stocker les données à servir via ce point d'accès multirégional. Vous pouvez utiliser Réplication entre régions (CRR) S3 pour synchroniser les données entre les compartiments de ces régions. Vous pourrez ensuite demander ou écrire des données via le point de terminaison global du point d'accès multi-régions. Amazon S3 répond automatiquement aux demandes de jeu de données répliquées à partir de la région la plus proche disponible. Les points d'accès multi-régions sont également compatibles avec les applications exécutées dans le cloud privé virtuel (VPC) Amazon, notamment celles qui utilisent AWS PrivateLink pour Amazon S3.

L'image suivante est une représentation graphique d'un point d'accès multi-régions Amazon S3 dans une configuration active-active. Le graphique montre comment les demandes Amazon S3 sont automatiquement acheminées vers les compartiments de la Région AWS active la plus proche.



L'image suivante est une représentation graphique d'un point d'accès multi-régions Amazon S3 dans une configuration active-passive. Le graphique montre comment vous pouvez contrôler le trafic d'accès aux données d'Amazon S3 pour basculer entre une Régions AWS active et passive.



Pour plus d'informations sur l'utilisation des points d'accès multi-régions, consultez Tutorial: Getting started with Amazon S3 Multi-Region Access Points (Didacticiel : Débuter à l'aide des points d'accès multi-régions Amazon S3).

Rubriques

- Création de points d'accès multi-Régions
- Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink
- Effectuer des requêtes par l'intermédiaire d'un point d'accès multi-régions

Création de points d'accès multi-Régions

Pour créer un point d'accès multi-régions Amazon S3, procédez comme suit :

- Spécifiez le nom du point d'accès multi-régions.
- Choisissez un compartiment dans chaque compartiment dans Région AWS lequel vous souhaitez répondre aux demandes du point d'accès multirégional.
- Configurez les paramètres de blocage de l'accès public Amazon S3 pour le point d'accès multi-régions.

Vous fournissez toutes cces informations dans une demande de création, qu'Amazon S3 traite de manière asynchrone. Amazon S3 fournit un jeton que vous pouvez utiliser pour surveiller l'état de la demande de création asynchrone.

Veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions provenant d' AWS Identity and Access Management Access Analyzer avant d'enregistrer votre stratégie. IAM Access Analyzer exécute des vérifications de politiques pour valider votre politique par rapport à la grammaire de politique et aux bonnes pratiques IAM. Ces vérifications génèrent des résultats et fournissent des recommandations exploitables pour vous aider à créer des stratégies fonctionnelles et conformes aux bonnes pratiques en matière de sécurité. Pour en savoir plus sur la validation des politiques à l'aide d'IAM Access Analyzer, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM. Pour afficher la liste des avertissements, erreurs et suggestions renvoyés par IAM Access Analyzer, consultez la Référence de vérification de stratégie IAM Access Analyzer.

Lorsque vous utilisez l'API, la demande de création d'un point d'accès multi-Régions est asynchrone. Lorsque vous soumettez une demande de création d'un point d'accès multi-Régions, Amazon S3 autorise la demande de manière synchrone. Il renvoie ensuite immédiatement un jeton que vous pouvez utiliser pour suivre la progression de la demande de création. Pour en savoir plus sur le suivi des demandes asynchrones pour créer et gérer des points d'accès multi-Régions, consultez Utilisation de points d'accès multi-régions avec des opérations d'API prises en charge.

Après avoir créé le point d'accès multi-Régions, vous pourrez créer une politique de contrôle d'accès pour celui-ci. Chaque point d'accès multi-Régions peut être associé à une politique. Une politique de point d'accès multi-régions est une politique basée sur les ressources qui vous permet de limiter l'utilisation du point d'accès multi-régions par ressource, utilisateur ou autres conditions.

> **Note**
>
> Pour qu'une application ou un utilisateur puisse accéder à un objet via un point d'accès multi-régions, les deux stratégies suivantes doivent autoriser la demande :
>
> - La stratégie d'accès associée au point d'accès multi-régions
> - La stratégie d'accès pour le compartiment sous-jacent qui contient l'objet
>
> Lorsque les deux politiques sont différentes, la politique la plus restrictive a la priorité. Pour simplifier la gestion des autorisations pour les points d'accès multi-régions, vous pouvez déléguer le contrôle d'accès du compartiment au point d'accès multi-régions. Pour plus

d'informations, consultez the section called "Exemples de politique de point d'accès multi-régions".

L'utilisation d'un compartiment avec un point d'accès multi-régions ne modifie pas le comportement du compartiment si le compartiment est accessible via le nom de compartiment existant ou un nom Amazon Resource Name (ARN). Toutes les opérations existantes sur le compartiment continuent de fonctionner comme auparavant. Les restrictions que vous incluez dans une politique de point d'accès s'appliquent uniquement aux demandes effectuées via ce point d'accès.

Vous pouvez mettre à jour la politique pour un point d'accès multi-Régions après l'avoir créée, mais vous ne pouvez pas la supprimer. Vous pouvez toutefois mettre à jour la politique de point d'accès multi-régions pour refuser toutes les autorisations.

Rubriques

Règles relatives à l'attribution de noms pour les points d'accès multi-Régions Amazon S3

Lorsque vous créez un point d'accès multi-Régions, vous lui attribuez un nom, qui est une chaîne que vous choisissez. Vous ne pourrez pas modifier le nom du point d'accès multi-Régions après sa création. Le nom doit être unique dans votre Compte AWS, et il doit être conforme aux exigences de dénomination répertoriées dans Restrictions et limitations des points d'accès multi-régions. Pour vous aider à identifier le point d'accès multi-Régions, utilisez un nom significatif pour vous, pour votre organisation ou qui reflète le scénario.

Vous utiliserez ce nom lors de l'appel d'opérations de gestion de points d'accès multi-Régions, telles que `GetMultiRegionAccessPoint` et `PutMultiRegionAccessPointPolicy`. Le nom n'est pas utilisé pour envoyer des demandes au point d'accès multi-régions, et il n'est pas nécessaire de l'exposer aux clients qui effectuent des demandes à l'aide du point d'accès multi-régions.

Quand Amazon S3 crée un point d'accès multi-Régions, il lui attribue automatiquement un alias. Cet alias est une chaîne alphanumérique unique qui se termine par `.mrap`. L'alias est utilisé pour créer le nom d'hôte et le nom Amazon Resource Name (ARN) d'un point d'accès multi-Régions. Le nom complet est également basé sur l'alias du point d'accès multi-Régions.

Vous ne pouvez pas déterminer le nom d'un point d'accès multi-Régions à partir de son alias, de sorte que vous pouvez divulguer un alias sans risque d'exposer le nom, l'objectif ou le propriétaire du point d'accès multi-Régions. Amazon S3 sélectionne l'alias pour chaque nouveau point d'accès multi-Régions et l'alias ne peut pas être modifié. Pour en savoir plus sur le traitement d'un point d'accès multi-Régions, consultez Effectuer des requêtes par l'intermédiaire d'un point d'accès multi-régions.

Les alias de points d'accès multi-Régions sont uniques dans le temps et ne sont pas basés sur le nom ou sur la configuration d'un point d'accès multi-Régions. Si vous créez un point d'accès multi-Régions, puis le supprimez et en créez un autre portant le même nom et ayant la même configuration, le deuxième point d'accès multi-Régions aura un alias différent du premier. Les nouveaux points d'accès multi-Régions ne peuvent jamais avoir le même alias qu'un point d'accès multi-Régions précédent.

Règles de sélection des compartiments pour les points d'accès multi-Régions Amazon S3

Chaque point d'accès multi-Régions est associé aux Régions où vous souhaitez traiter les demandes. Le point d'accès multi-Régions doit être associé à exactement un compartiment dans chacune de ces Régions. Vous indiquez le nom de chaque compartiment dans la demande de création du point d'accès multi-Régions. Les compartiments qui prennent en charge le point d'accès multirégional peuvent se trouver soit dans le même Compte AWS que celui qui possède le point d'accès multirégional, soit dans un autre. Comptes AWS

Un seul compartiment peut être utilisé par plusieurs points d'accès multi-Régions.

> ⚠ Important
>
> - Vous pouvez indiquer les compartiments associés à un point d'accès multi-Régions uniquement au moment de sa création. Une fois créé, vous ne pourrez pas ajouter, modifier ou supprimer des compartiments de la configuration de point d'accès multi-Régions. Pour modifier les compartiments, vous devrez supprimer tout le point d'accès multi-Régions et en créer un autre.

- Vous ne pouvez pas supprimer un compartiment qui fait partie d'un point d'accès multi-Régions. Si vous souhaitez supprimer un compartiment attaché à un point d'accès multi-régions, supprimez d'abord le point d'accès multi-régions.
- Si vous ajoutez un compartiment qui appartient à un autre compte à votre point d'accès multi-régions, le propriétaire doit également mettre à jour sa politique de compartiment pour accorder des autorisations sur le point d'accès multi-régions. Sinon, le point d'accès multi-régions ne sera pas en mesure de récupérer les données de ce compartiment. Pour des exemples de politiques indiquant comment accorder un tel accès, consultez Exemples de politique de point d'accès multi-régions.
- Les Régions ne prennent pas toutes en charge les points d'accès multi-Régions. Pour obtenir la liste des Régions de prise en charge, consultez Restrictions et limitations des points d'accès multi-régions.

Vous pouvez créer des règles de réplication pour synchroniser les données entre les compartiments. Ces règles vous permettent de copier automatiquement les données des compartiments source vers des compartiments de destination. Le fait d'avoir des compartiments connectés à un point d'accès multi-Régions n'affecte pas le fonctionnement de la réplication. La configuration de la réplication avec des points d'accès multi-Régions est décrite dans une section ultérieure.

Important

Lorsque vous effectuez une demande à un point d'accès multi-régions, le point d'accès multi-régions ne connaît pas le contenu des compartiments dans le point d'accès multi-régions. Par conséquent, le compartiment qui reçoit la demande peut ne pas contenir les données demandées. Pour créer des jeux de données cohérents dans les compartiments Amazon S3 associés à un point d'accès multi-régions, nous vous recommandons de configurer la réplication entre régions (CRR) S3. Pour plus d'informations, consultez Configuration de la réplication à utiliser avec des points d'accès multi-régions.

Création d'un point d'accès multi-régions Amazon S3

L'exemple suivant montre comment créer un point d'accès multi-régions à l'aide de la console Amazon S3.

Utilisation de la console S3

Créer un point d'accès multi-Régions

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Choisissez Créer des points d'accès multi-régions pour commencer à créer votre point d'accès multi-régions.
4. Sur la page Point d'accès multi-régions, indiquez un nom pour le point d'accès multi-régions dans le champ Nom du point d'accès multi-régions.
5. Sélectionnez les compartiments qui seront associés à ce point d'accès multi-régions. Vous pouvez choisir des compartiments qui se trouvent dans votre compte ou choisir des compartiments provenant d'autres comptes.

> **Note**
>
> Vous devez ajouter au moins un compartiment à partir de votre compte ou d'autres comptes. Sachez également que les points d'accès multi-régions ne prennent en charge qu'un seul compartiment par Région AWS. Par conséquent, vous ne pouvez pas ajouter deux compartiments de la même région. Les Régions AWS désactivées par défaut ne sont pas prises en charge.

- Pour ajouter un compartiment qui se trouve dans votre compte, choisissez Ajouter des compartiments. Une liste de tous les compartiments de votre compte s'affiche. Vous pouvez rechercher votre compartiment par nom ou trier les noms des compartiments par ordre alphabétique.
- Pour ajouter un compartiment à partir d'un autre compte, choisissez Ajouter un compartiment à partir d'autres comptes. Assurez-vous de connaître le nom et l' Compte AWS identifiant exacts du compartiment, car vous ne pouvez pas rechercher ou rechercher des compartiments dans d'autres comptes.

> **Note**
>
> Vous devez saisir un Compte AWS identifiant et un nom de compartiment valides. Le compartiment doit également se trouver dans une région prise en charge, sinon vous rencontrerez une erreur lorsque vous tenterez de créer votre point d'accès multi-régions. Pour obtenir la liste des régions qui prennent en charge les points d'accès multi-régions, consultez Restrictions et limitations des points d'accès multi-régions.

6. (Facultatif) Si vous devez supprimer un compartiment que vous avez ajouté, choisissez Supprimer.

> **Note**
>
> Vous ne pouvez pas ajouter ni supprimer de compartiments à ce point d'accès multi-régions une fois que vous avez fini de le créer.

7. Sous Paramètres de blocage de l'accès public à ce point d'accès multi-Régions, sélectionnez les paramètres de blocage de l'accès public que vous souhaitez appliquer au point d'accès multi-Régions. Tous les paramètres de blocage de l'accès public sont activés par défaut pour les points d'accès. Nous vous recommandons de laisser tous les paramètres activés, sauf si vous savez que vous avez un besoin spécifique de désactiver l'un d'entre eux.

> **Note**
>
> Vous ne pouvez pas modifier les paramètres de blocage de l'accès public après la création du point d'accès multi-régions. Par conséquent, si vous avez l'intention de bloquer l'accès public, assurez-vous que vos applications fonctionnent correctement sans accès public avant de créer un point d'accès multi-régions.

8. Choisissez Créer un point d'accès multi-Régions.

> **Important**
>
> Quand vous ajoutez un compartiment qui appartient à un autre compte à votre point d'accès multi-régions, le propriétaire doit également mettre à jour sa politique de compartiment pour accorder des autorisations sur le point d'accès multi-régions. Sinon, le point d'accès multi-régions ne sera pas en mesure de récupérer les données de ce compartiment. Pour des

exemples de politiques indiquant comment accorder un tel accès, consultez Exemples de politique de point d'accès multi-régions.

À l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour créer un point d'accès multirégional. Lorsque vous créez le point d'accès multi-régions, vous devez indiquer tous les compartiments qu'il devra prendre en charge. Vous ne pouvez pas ajouter des compartiments au point d'accès multi-régions après sa création.

L'exemple suivant crée un point d'accès multi-régions avec deux compartiments à l'aide de l' AWS CLI. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{
        "Name": "simple-multiregionaccesspoint-with-two-regions",
        "PublicAccessBlock": {
            "BlockPublicAcls": true,
            "IgnorePublicAcls": true,
            "BlockPublicPolicy": true,
            "RestrictPublicBuckets": true
        },
        "Regions": [
            { "Bucket": "example-s3-bucket1" },
            { "Bucket": "example-s3-bucket2" }
        ]
    }' --region us-west-2
```

Bloquer l'accès public à l'aide des points d'accès multi-Régions Amazon S3

Chaque point d'accès multi-Régions dispose de paramètres distincts pour bloquer l’accès public Amazon S3. Ces paramètres fonctionnent conjointement avec les paramètres de blocage de l'accès public pour le Compte AWS propriétaire du point d'accès multirégional et des compartiments sous-jacents.

Quand Amazon S3 autorise une demande, il applique la combinaison la plus restrictive de ces paramètres. Si les paramètres de blocage de l'accès public pour l'une de ces ressources (le compte propriétaire du point d'accès multi-régions, le compartiment sous-jacent ou le compte propriétaire

du compartiment) bloquent l'accès à l'action ou à la ressource demandée, Amazon S3 rejettera la demande.

Nous vous recommandons d'activer tous les paramètres de blocage de l'accès public sauf si vous devez spécifiquement en désactiver certains. Tous les paramètres de blocage de l'accès public sont activés par défaut pour les points d'accès. Si l'option de blocage de l'accès public est activée, le point d'accès multi-régions ne peut pas accepter les demandes basées sur Internet.

> **Important**
>
> Vous ne pouvez pas modifier les paramètres de blocage de l'accès public après la création du point d'accès multi-régions.

Pour en savoir plus sur le blocage de l'accès public Amazon S3, consultez Blocage de l'accès public à votre stockage Amazon S3.

Affichage des détails de la configuration des points d'accès multi-régions Amazon S3

L'exemple suivant montre comment afficher les détails de la configuration d'un point d'accès multi-régions à l'aide de la console Amazon S3.

Utilisation de la console S3

Créer un point d'accès multi-Régions

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Choisissez le nom du point d'accès multi-régions dont vous souhaitez afficher les détails de la configuration.

   - L'onglet Propriétés répertorie tous les compartiments associés à votre point d'accès multi-régions, la date de création, l'Amazon Resource Name (ARN) et l'alias. La colonne ID de Compte AWS répertorie également tous les compartiments appartenant à des comptes externes associés à votre point d'accès multi-régions.

- L'onglet Autorisations répertorie les paramètres de blocage d'accès public appliqués aux compartiments associés à ce point d'accès multi-régions. Vous pouvez également afficher la politique de point d'accès multi-régions, si vous en avez créé une. L'alerte Infos sur la page Autorisations répertorie également tous les compartiments (de votre compte et d'autres comptes) pour ce point d'accès multi-régions pour lesquels le paramètre L'accès public est bloqué est activé.
- L'onglet Réplication et basculement fournit une vue cartographique des compartiments associés à votre point d'accès multi-régions et des régions dans lesquelles se trouvent les compartiments. S'il existe des compartiments provenant d'un autre compte dont vous n'êtes pas autorisé à extraire des données, la région sera marquée en rouge sur la carte Récapitulatif de réplication, ce qui indique qu'il s'agit d'une Région AWS présentant des erreurs lors de l'obtention du statut de réplication.

  > **Note**
  >
  > Pour récupérer les informations sur le statut de réplication d'un compartiment dans un compte externe, le propriétaire du compartiment doit vous accorder l'autorisation `s3:GetBucketReplication` conformément à sa politique en matière de compartiment.

  Cet onglet fournit également les métriques de réplication, les règles de réplication et les statuts de basculement pour les régions utilisées avec votre point d'accès multi-régions.

À l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour afficher les détails de configuration d'un point d'accès multirégional.

L' AWS CLI exemple suivant montre la configuration actuelle de votre point d'accès multirégional. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name example-s3-bucket1
```

Suppression d'un point d'accès multi-régions

La procédure suivante montre comment supprimer un point d'accès multi-régions à l'aide de la console Amazon S3.

La suppression d'un point d'accès multi-régions ne supprime pas les compartiments associés au point d'accès multi-régions, mais uniquement le point d'accès multi-régions.

Utilisation de la console S3

Pour supprimer un point d'accès multi-régions

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Sélectionnez le bouton d'option en regard du nom de votre point d'accès multi-régions.
4. Sélectionnez Delete (Supprimer).
5. Dans la boîte de dialogue Supprimer le point d'accès multirégional, entrez le nom du AWS compartiment que vous souhaitez supprimer.

   > **Note**
   >
   > Assurez-vous de saisir un nom de compartiment valide. Dans le cas contraire, le bouton Supprimer sera désactivé.

6. Choisissez Supprimer pour confirmer la suppression de votre point d'accès multi-régions.

À l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour supprimer un point d'accès multirégional. L'action ne supprime pas les compartiments associés au point d'accès multi-régions, mais uniquement le point d'accès multi-régions. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
 Name=example-multi-region-access-point-name
```

Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink

Vous pouvez utiliser les points d'accès multi-régions pour acheminer le trafic de demandes Amazon S3 entre plusieurs Régions AWS. Chaque point d'accès multi-régions achemine le trafic de demandes de données Amazon S3 depuis plusieurs sources sans que vous ayez à créer des configurations réseau complexes avec des points de terminaison distincts. Ces sources de trafic de demandes de données incluent :

- Trafic provenant d'un cloud privé virtuel (VPC)
- Trafic provenant de centres de données sur site vers AWS PrivateLink
- Trafic provenant de l'Internet public

Si vous établissez une connexion AWS PrivateLink à un point d'accès multi-régions S3, vous pouvez acheminer les demandes S3 vers AWS ou entre plusieurs Régions AWS via une connexion privée en utilisant une architecture et une configuration réseau simples. Lorsque vous utilisez AWS PrivateLink, vous n'avez pas besoin de configurer une connexion d'appairage de VPC.

Rubriques

Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink

AWS PrivateLink vous fournit une connectivité privée à Amazon S3 à l'aide d'adresses IP privées dans votre Virtual Private Cloud (VPC). Vous pouvez provisionner un ou plusieurs points de terminaison d'interface dans votre VPC pour vous connecter aux points d'accès multi-Régions Amazon S3.

Vous pouvez créer des points de terminaison com.amazonaws.s3-global.accesspointpour les points d'accès multi-Régions via le AWS Management Console, AWS CLI, ou les SDK AWS. Pour en savoir plus sur la configuration d'un point de terminaison d'interface pour un point d'accès multi-Régions, consultez Points de terminaison d'un VPC d'interface dans le guide de l'utilisateur VPC.

Pour adresser des demandes à un point d'accès multi-Régions via des points de terminaison d'interface, procédez comme suit pour configurer le VPC et le point d'accès multi-Régions.

Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink

1. Créez ou disposez d'un point de terminaison d'un VPC approprié qui peut se connecter à des points d'accès multi-Régions. Pour en savoir plus sur les points de terminaison d'un VPC, consultez Points de terminaison d'un VPC d'interface dans le guide de l'utilisateur VPC.

   > **Important**
   >
   > Veillez à créer un point de terminaison com.amazonaws.s3-global.accesspoint. Les autres types de points de terminaison ne peuvent pas accéder aux points d'accès multi-Régions.

   Après la création de ce point de terminaison d'un VPC, toutes les demandes de point d'accès multi-Régions dans le VPC seront acheminées via ce point de terminaison si vous disposez d'un DNS privé activé pour le point de terminaison. Cela est activé par défaut.
2. Si la politique de point d'accès multi-Régions ne prend pas en charge les connexions à partir des points de terminaison d'un VPC, vous devrez la mettre à jour.
3. Vérifiez que les politiques de compartiment individuelles autorisent l'accès aux utilisateurs du point d'accès multi-Régions.

N'oubliez pas que les points d'accès multi-Régions fonctionnent en acheminant les demandes vers des compartiments, non en les exécutant eux-mêmes. Il est important de s'en souvenir, car l'expéditeur de la demande doit disposer des autorisations sur le point d'accès multi-Régions et être autorisé à accéder aux compartiments individuels dans le point d'accès multi-Régions. Sinon, la demande risque d'être acheminée vers un compartiment où l'expéditeur n'a pas les autorisations nécessaires pour répondre à la demande. Un point d'accès multi-régions et les compartiments qui y sont associés peuvent appartenir au même compte AWS ou à un autre. Toutefois, les VPC de comptes différents peuvent utiliser un point d'accès multi-Régions si les autorisations sont configurées correctement.

Pour cette raison, la politique de point de terminaison d'un VPC doit autoriser l'accès au point d'accès multi-Régions et à chaque compartiment sous-jacent qui doit être capable de traiter les demandes. Par exemple, supposons que vous ayez un point d'accès multi-Régions avec l'alias

mfzwi23gnjvgw.mrap. Il est soutenu par des compartiments DOC-EXAMPLE-BUCKET1 et DOC-EXAMPLE-BUCKET2 qui appartiennent tous deux à un compte AWS123456789012. Dans ce cas, la politique de point de terminaison VPC suivante permettrait que les demandes GetObject du VPC faites à mfzwi23gnjvgw.mrap soient satisfaites par l'un ou l'autre des compartiments de sauvegarde.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Read-buckets-and-MRAP-VPCE-policy",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*",
            "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
        ]
    }]
}
```

Comme mentionné précédemment, vous devez également vous assurer que la politique de point d'accès multi-Régions est configurée de manière à prendre en charge l'accès via un point de terminaison d'un VPC. Il n'est pas nécessaire de préciser le point de terminaison d'un VPC qui demande l'accès. L'exemple de politique suivant accorderait l'accès à tout demandeur essayant d'utiliser le point d'accès multi-régions pour les demandes GetObject.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Open-read-MRAP-policy",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
          ],
        "Resource": "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
```

```
    }]
}
```

Et, bien entendu, les compartiments individuels auraient chacun besoin d'une politique pour prendre en charge l'accès à partir des demandes soumises via le point de terminaison d'un VPC. L'exemple de politique suivant accorde l'accès en lecture à tous les utilisateurs anonymes, ce qui inclut les demandes effectuées via le point de terminaison VPC.

```
{
    "Version":"2012-10-17",
   "Statement": [
   {
       "Sid": "Public-read",
       "Effect":"Allow",
       "Principal": "*",
       "Action": "s3:GetObject",
       "Resource": [
           "arn:aws:s3:::DOC-EXAMPLE-BUCKET1",
           "arn:aws:s3:::DOC-EXAMPLE-BUCKET2/*"]
    }]
}
```

Pour en savoir plus sur la modification d'une politique de point de terminaison d'un VPC, consultez Contrôle de l'accès aux services avec des points de terminaison d'un VPC dans le guide de l'utilisateur VPC.

Supprimer l'accès à un point d'accès multi-Régions à partir d'un point de terminaison d'un VPC

Si vous possédez un point d'accès multi-régions et souhaitez en supprimer l'accès à partir d'un point de terminaison d'interface, vous devrez fournir une nouvelle stratégie d'accès au point d'accès multi-régions qui empêche l'accès aux demandes provenant des points de terminaison d'un VPC. Cependant, si les compartiments de votre point d'accès multi-régions prennent en charge les demandes via des points de terminaison d'un VPC, ils continueront à prendre en charge ces demandes. Si vous souhaitez empêcher cette prise en charge, vous devrez également mettre à jour les politiques des compartiments. La fourniture d'une nouvelle stratégie d'accès au point d'accès multi-régions empêche uniquement l'accès au point d'accès multi-régions, pas aux compartiments sous-jacents.

> **Note**
>
> Vous ne pouvez pas supprimer une politique d'accès pour un point d'accès multi-Régions. Pour supprimer l'accès à un point d'accès multi-Régions, vous devez fournir une nouvelle politique d'accès avec l'accès modifié souhaité.

Au lieu de mettre à jour la stratégie d'accès pour le point d'accès multi-régions, vous pouvez mettre à jour les politiques de compartiment pour empêcher les demandes via les points de terminaison d'un VPC. Dans ce cas, les utilisateurs peuvent toujours accéder au point d'accès multi-régions via le point de terminaison d'un VPC. Mais si la demande de point d'accès multi-régions est acheminée vers un compartiment où la politique de compartiment empêche l'accès, elle générera un message d'erreur.

Effectuer des requêtes par l'intermédiaire d'un point d'accès multi-régions

Comme les autres ressources, les points d'accès multi-régions Amazon S3 comportent des noms Amazon Resource Name (ARN). Vous pouvez utiliser ces ARN pour diriger les demandes vers les points d'accès multi-régions en utilisant l'AWS Command Line Interface (AWS CLI), les kits SDK AWS ou l'API Amazon S3. Vous pouvez également utiliser ces ARN pour identifier les points d'accès multi-régions dans les stratégies de contrôle d'accès. Un ARN de point d'accès multi-régions n'inclut pas ou ne divulgue pas son nom. Pour de plus amples informations sur l'utilisation des ARN, veuillez consulter Amazon Resource Names (ARN) dans le Références générales AWS.

> **Note**
>
> L'alias et l'ARN du point d'accès multirégion ne peuvent pas être utilisés de manière interchangeable.

Les ARN des points d'accès multi-régions utilisent le format suivant :

`arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*

Voici quelques exemples d'ARN de points d'accès multi-régions :

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` représente le point d'accès multi-régions avec l'alias `mfzwi23gnjvgw.mrap`, détenu par le Compte AWS `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/*` représente tous les points d'accès multi-régions du compte `123456789012`. Cet ARN correspond à tous les points d'accès multi-régions du compte `123456789012`, mais il ne correspond à aucun point d'accès Amazon S3 régional, car l'ARN n'inclut pas de Région AWS. En revanche, l'ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` correspond à tous les points d'accès Amazon S3 régionaux de la région `us-west-2` pour le compte `123456789012`, mais il ne correspond à aucun point d'accès multi-régions.

Les ARN des objets accessibles via un point d'accès multi-régions utilisent le format suivant :

`arn:aws:s3::`*`account_id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*`//`*`key`*

Comme pour les ARN des points d'accès multi-régions, les ARN des objets accessibles via des points d'accès multi-régions n'incluent pas la Région AWS. Voici quelques exemples.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` représente l'objet `-01`, accessible via le point d'accès multi-régions avec l'alias `mfzwi23gnjvgw.mrap`, qui appartient à un compte `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` représente tous les objets accessibles via le point d'accès multi-régions avec l'alias `mfzwi23gnjvgw.mrap`, dans un compte `123456789012`.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` représente tous les objets accessibles sous le préfixe `-01/finance/` pour le point d'accès multi-régions avec l'alias `mfzwi23gnjvgw.mrap`, dans un compte `123456789012`.

Noms d'hôte des points d'accès multi-Régions

Vous pouvez accéder aux données Amazon S3 via un point d'accès multi-régions à l'aide du nom d'hôte du point d'accès multi-régions. Les requêtes peuvent être dirigées vers ce nom d'hôte depuis l'Internet public. Si vous avez configuré une ou plusieurs passerelles Internet pour le point d'accès multi-régions, les requêtes peuvent également être dirigées vers ce nom d'hôte depuis un cloud privé virtuel (VPC). Pour en savoir plus sur la création de points de terminaison d'interface VPC à utiliser

avec des points d'accès multi-Régions, consultez Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink.

Pour effectuer des demandes via un point d'accès multi-régions à partir d'un VPC utilisant un point de terminaison d'un VPC, vous pouvez utiliser AWS PrivateLink. Lorsque vous effectuez des demandes auprès d'un point d'accès multi-régions en utilisant AWS PrivateLink, vous ne pouvez pas utiliser directement un nom de système de nom de domaine (DNS) régional spécifique au point de terminaison qui se termine par *region*.vpce.amazonaws.com. Ce nom d'hôte ne sera pas associé à un certificat, il ne peut donc pas être utilisé directement. Vous pouvez toujours utiliser le nom de système de nom de domaine (DNS) public du point de terminaison d'un VPC comme une cible CNAME ou ALIAS. Vous pouvez également activer le système de nom de domaine (DNS) privé sur le point de terminaison et utiliser le nom du système de nom de domaine (DNS) standard *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com du point d'accès multi-régions, comme décrit dans cette section.

Lorsque vous faites des demandes à l'API pour les opérations sur les données Amazon S3 (par exemple, GetObject) via un point d'accès multi-régions, le nom d'hôte de la demande est le suivant :

MultiRegionAccessPoint_alias.accesspoint.s3-global.amazonaws.com

Par exemple, pour effectuer une requête GetObject via le point d'accès multi-régions avec l'alias mfzwi23gnjvgw.mrap, faites une demande au nom d'hôte mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com. La partie s3-global du nom d'hôte indique que ce nom d'hôte n'est pas destiné à une région particulière.

Faire des demandes via un point d'accès multi-Régions est similaire à faire des demandes via un point d'accès à une seule Région. Toutefois, il est important de connaître les différences suivantes :

- Les ARN des points d'accès multi-régions n'incluent aucune Région AWS. Ils suivent le format arn:aws:s3::*account-id*:accesspoint/*MultiRegionAccessPoint_alias*.
- Pour les demandes effectuées via des opérations d'API (ces demandes ne nécessitent pas l'utilisation d'un ARN), les points d'accès multi-régions utilisent un modèle de point de terminaison différent. Le est est *MultiRegionAccessPoint_alias*.accesspoint.s3-global.amazonaws.com (par exemple, mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com). Notez les différences par rapport à un point d'accès à une seule Région :
  - Les noms d'hôtes de points d'accès multi-Régions utilisent leurs alias, pas le nom du point d'accès multi-Régions.

  - Les noms d'hôtes des points d'accès multi-régions n'incluent pas l'ID Compte AWS du propriétaire.
  - Les noms d'hôtes des points d'accès multi-régions n'incluent aucune Région AWS.
  - Les noms d'hôte des points d'accès multi-Régions incluent `s3-global.amazonaws.com` au lieu de `s3.amazonaws.com`.
- Les requêtes de points d'accès multi-régions doivent être signées en utilisant la version 4A de la signature (SigV4A). Si vous utilisez le kit SDK AWS, il convertira automatiquement une SigV4 en SigV4A. Par conséquent, vérifiez que votre version de votre kit SDK AWS prend en charge SigV4A comme mise en œuvre de la signature qui est utilisée pour signer les demandes pour les Région AWS mondiales. Pour plus d'informations sur SigV4A, consultez Signature des demandes d'API AWS dans la Références générales AWS.

Points d'accès multi-Régions et Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration est une fonction qui permet des transferts de données rapides vers des compartiments. L'accélération du transfert est configurée au niveau de chaque compartiment. Pour de plus amples informations sur Transfer Acceleration, veuillez consulter Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.

Les points d'accès multi-régions utilisent un mécanisme de transfert accéléré similaire à l'accélération du transfert pour envoyer des objets volumineux sur le réseau AWS. Pour cette raison, vous n'avez pas besoin d'utiliser l'accélération du transfert lorsque vous envoyez des requêtes via un point d'accès multi-régions. Ces performances de transfert accrues sont automatiquement intégrées au point d'accès multi-régions.

Rubriques

- Autorisations
- Restrictions et limitations des points d'accès multi-régions
- Routage des demandes de points d'accès multi-Régions
- Contrôles de basculement des points d'accès multi-régions Amazon S3
- Configuration de la réplication à utiliser avec des points d'accès multi-régions
- Utilisation de points d'accès multi-régions avec des opérations d'API prises en charge
- Surveillance et journalisation des demandes effectuées via un point d'accès multi-Régions vers les ressources sous-jacentes

Autorisations

Les points d'accès multi-régions Amazon S3 peuvent simplifier l'accès aux données pour les compartiments Amazon S3 dans plusieurs Régions AWS. Les points d'accès multi-régions sont des points de terminaison mondiaux nommés que vous pouvez utiliser pour effectuer des opérations sur les objets d'accès aux données Amazon S3, telles que `GetObject` et `PutObject`. Chaque point d'accès multi-régions peut comporter des autorisations et des contrôles de réseau distincts pour toute requête effectuée par le point de terminaison global.

Chaque point d'accès multi-régions peut également appliquer une stratégie d'accès personnalisée qui fonctionne conjointement avec la politique de compartiment attachée au compartiment sous-jacent. Pour qu'une demande aboutisse, tous les éléments suivants doivent autoriser l'opération :

- La politique de point d'accès multi-régions
- La politique AWS Identity and Access Management (IAM) sous-jacente
- La politique de compartiment sous-jacente (vers laquelle la requête est routée).

Vous pouvez configurer toute politique de point d'accès multi-régions pour qu'elle n'accepte que les requêtes provenant d'utilisateurs ou de groupes IAM spécifiques. Pour obtenir un exemple de la manière de procéder, consultez l'exemple 2 dans the section called “Exemples de politique de point d'accès multi-régions”. Pour limiter l'accès aux données Amazon S3 à un réseau privé, vous pouvez configurer la politique de point d'accès multi-régions pour accepter les requêtes provenant uniquement d'un cloud privé virtuel (VPC).

Par exemple, supposons que vous créez une requête `GetObject` via un point d'accès multi-Régions à l'aide d'un utilisateur appelé `AppDataReader` dans votre compte AWS. Pour vous assurer que la demande ne sera pas refusée, l'utilisateur `AppDataReader` doit se voir accorder l'autorisation `s3:GetObject` par le point d'accès multi-Régions et par chaque compartiment sous-jacent au point d'accès multi-Régions. `AppDataReader` ne sera pas en mesure de récupérer des données d'un compartiment qui n'octroie pas cette autorisation.

> **Important**
>
> La délégation du contrôle d'accès d'un compartiment à une politique de point d'accès multi-régions ne modifie pas le comportement du compartiment lorsqu'on y accède directement par son nom de compartiment ou Amazon Resource Name (ARN). Toutes les opérations effectuées directement sur le compartiment continueront à fonctionner comme avant. Les

restrictions que vous incluez dans une politique de point d'accès multi-régions s'appliquent uniquement aux demandes effectuées via ce point d'accès.

Gestion de l'accès public vers un point d'accès multi-Régions

Les points d'accès multi-Régions prennent en charge des paramètres de blocage d'accès public indépendants pour chaque point d'accès multi-Régions. Lorsque vous créez un point d'accès multi-Régions, vous pouvez indiquer les paramètres de blocage d'accès public qui s'appliquent à ce point d'accès multi-Régions.

> **Note**
>
> Tous les paramètres de blocage de l'accès public qui sont activés dans Paramètres de blocage de l'accès public pour ce compte (dans votre propre compte) ou dans Paramètres de blocage public pour les compartiments externes s'appliquent, même si les paramètres indépendants de blocage de l'accès public pour votre point d'accès multi-régions sont désactivés.

Pour toute requête effectuée par le biais d'un point d'accès multi-régions, Amazon S3 évalue les paramètres de blocage de l'accès public pour les éléments suivants :

- Le point d'accès multi-régions
- Les compartiments sous-jacents (y compris les compartiments externes)
- Le compte propriétaire du point d'accès multi-régions
- Le compte propriétaire des compartiments sous-jacents (y compris les comptes externes)

Si l'un de ces paramètres indique que la demande doit être bloquée, Amazon S3 rejettera la demande. Pour en savoir de plus sur le blocage de l'accès public Amazon S3, consultez Blocage de l'accès public à votre stockage Amazon S3.

> **Important**
>
> Tous les paramètres de blocage de l'accès public sont activés par défaut pour les points d'accès. Vous devez désactiver explicitement tous les paramètres que vous ne souhaitez pas appliquer à un point d'accès multi-Régions.

Vous ne pouvez pas modifier les paramètres de blocage de l'accès public après la création du point d'accès multi-régions.

Affichage des paramètres de blocage d'accès public pour un point d'accès multi-régions

Pour afficher les paramètres de blocage d'accès public d'un point d'accès multi-régions, procédez comme suit

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Choisissez le nom du point d'accès multi-régions que vous voulez examiner.
4. Choisissez l'onglet Permissions (Autorisations).
5. Sous Block Public Access settings for this Multi-Region Access Point (Bloquer les paramètres d'accès public pour ce point d'accès multi-régions), vérifiez les paramètres de blocage de l'accès public pour votre point d'accès multi-régions.

> **Note**
>
> Vous ne pouvez pas modifier les paramètres de blocage de l'accès public après la création du point d'accès multi-régions. Par conséquent, si vous avez l'intention de bloquer l'accès public, assurez-vous que vos applications fonctionnent correctement sans accès public avant de créer un point d'accès multi-régions.

Utilisation d'une politique de point d'accès multi-régions

L'exemple suivant de politique de point d'accès multi-régions accorde à un utilisateur IAM l'accès à la liste et au téléchargement de fichiers depuis votre point d'accès multi-régions. Pour utiliser cet exemple de politique, remplacez *user input placeholders* par vos propres informations.

```
{
  "Version":"2012-10-17",
```

```
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
         },
         "Action":[
            "s3:ListBucket",
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
            "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/
*"
         ]
      }
   ]
}
```

Pour associer votre politique de point d'accès multi-régions au point d'accès multi-régions spécifié avec AWS Command Line Interface (AWS CLI), utilisez la commande `put-multi-region-access-point-policy` suivante. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations. Chaque point d'accès multi-régions ne peut avoir qu'une seule politique, donc une demande faite à l'action `put-multi-region-access-point-policy` remplace toute politique existante associée au point d'accès multi-régions spécifié.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint",
 "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::111122223333:root
\" }, \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
 [ \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
 \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
\" ] } }" }
```

Pour interroger les résultats de l'opération précédente, utilisez la commande suivante :

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Pour récupérer votre politique de point d'accès multi-régions, utilisez la commande suivante :

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=DOC-EXAMPLE-BUCKET-MultiRegionAccessPoint
```

Modification de la politique de point d'accès multi-régions

La politique de point d'accès multi-régions (écrite en JSON) fournit un accès de stockage aux compartiments Amazon S3 qui sont utilisés avec ce point d'accès multi-régions. Vous pouvez autoriser ou empêcher des principaux spécifiques d'effectuer diverses actions sur votre point d'accès multi-régions. Lorsqu'une requête est routée vers un compartiment via le point d'accès multi-régions, les stratégies d'accès du point d'accès multi-régions et du compartiment s'appliquent. La stratégie d'accès la plus restrictive a toujours la priorité.

> **Note**
>
> Si un compartiment contient des objets appartenant à d'autres comptes, la politique du point d'accès multi-régions ne s'applique pas aux objets appartenant à d'autres Comptes AWS.

Après avoir appliqué une politique de point d'accès multi-régions, la politique ne peut pas être supprimée. Vous pouvez soit modifier la politique, soit créer une nouvelle politique qui écrase la politique existante.

Pour modifier la politique de point d'accès multi-régions

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Choisissez le nom du point d'accès multi-régions pour lequel vous souhaitez modifier la politique.
4. Choisissez l'onglet Permissions (Autorisations).
5. Faites défiler vers le bas jusqu'à la section Multi-Region Access Point policy (Politique de point d'accès multi-régions). Choisissez Edit (Modifier) pour mettre à jour la politique (en JSON).
6. La page Edit Multi-Region Access Point policy (Modifier la politique de point d'accès multi-régions) s'affiche. Vous pouvez soit saisir la politique directement dans le champ de texte, soit choisir Add statement (Ajouter une instruction) pour sélectionner les éléments de la politique dans une liste déroulante.

> **Note**
>
> La console affiche automatiquement l'Amazon Resource Name (ARN) du point d'accès multi-régions, que vous pouvez utiliser dans la politique. Pour obtenir des exemples de politique de point d'accès multi-régions, consultez the section called “Exemples de politique de point d'accès multi-régions”.

Exemples de politique de point d'accès multi-régions

Les points d'accès multi-régions Amazon S3 prennent en charge les politiques de ressources AWS Identity and Access Management (IAM). Ces politiques peuvent contrôler l'utilisation du point d'accès multi-régions par ressource, par utilisateur ou d'autres conditions. Pour qu'une application ou un utilisateur puisse accéder à des objets par le biais d'un point d'accès multi-régions, le point d'accès multi-régions et le compartiment sous-jacent doivent tous deux permettre le même accès.

Pour autoriser le même accès à la fois au point d'accès multi-régions et au compartiment sous-jacent, effectuez l'une des opérations suivantes :

- (Recommandé) Pour simplifier les contrôles d'accès lors de l'utilisation d'un point d'accès multi-régions Amazon S3, déléguez le contrôle d'accès pour le compartiment Amazon S3 au point d'accès multi-régions. Pour obtenir un exemple de la manière de procéder, reportez-vous à l'exemple 1 de cette section.
- Ajoutez les mêmes autorisations contenues dans la politique de point d'accès multi-régions à la politique de compartiment sous-jacente.

> **Important**
>
> La délégation du contrôle d'accès d'un compartiment à une politique de point d'accès multi-régions ne modifie pas le comportement du compartiment lorsqu'on y accède directement par son nom de compartiment ou Amazon Resource Name (ARN). Toutes les opérations effectuées directement sur le compartiment continueront à fonctionner comme avant. Les restrictions que vous incluez dans une politique de point d'accès multi-régions s'appliquent uniquement aux demandes effectuées via ce point d'accès.

Example 1 : délégation de l'accès à des points d'accès multi-régions spécifiques dans votre politique de compartiment (pour le même compte ou intercompte)

L'exemple suivant de politique de compartiment permet un accès complet à un point d'accès multi-régions spécifique. Cela signifie que tout accès à ce compartiment est contrôlé par les politiques qui sont attachées au point d'accès multi-régions. Nous vous recommandons de configurer vos compartiments de cette façon pour tous les cas d'utilisation qui ne demandent pas d'accès direct au compartiment. Vous pouvez utiliser cette structure de politique de compartiment pour les points d'accès multi-régions du même compte ou d'un autre compte.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
        }
    }]
}
```

> **Note**
>
> Si vous accordez l'accès à plusieurs points d'accès multi-régions, assurez-vous de lister chaque point d'accès multi-régions.

Example 2 : accorder l'accès d'un compte à un point d'accès multi-régions dans votre politique de point d'accès multi-régions

La stratégie de point d'accès multi-région suivante accorde l'autorisation au compte *123456789012* de lister et de lire les objets contenus dans le point d'accès multi-région défini par *MultiRegionAccessPoint_ARN*.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
       "Effect": "Allow",
       "Principal": {
          "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
       },
       "Action":[
          "s3:ListBucket",
          "s3:GetObject"
       ],
       "Resource":[
          "MultiRegionAccessPoint_ARN",
          "MultiRegionAccessPoint_ARN/object/*"
       ]
     }
  ]
}
```

Example 3 – Stratégie de point d'accès multi-région autorisant le listage des compartiments

La stratégie de point d'accès multi-région suivante accorde l'autorisation au compte *123456789012* de lister les objets contenus dans le point d'accès multi-région défini par *MultiRegionAccessPoint_ARN*.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action": "s3:ListBucket",
```

```
        "Resource": "MultiRegionAccessPoint_ARN"
      }
    ]
}
```

Restrictions et limitations des points d'accès multi-régions

Les points d'accès Amazon S3 comportent les restrictions et limitations suivantes :

- Noms des points d'accès multi-régions :
  - Doit être unique au sein d'un seul AWS compte
  - Ils doivent commencer par un chiffre ou une lettre minuscule.
  - Ils doivent comporter entre 3 et 50 caractères.
  - Ils ne peuvent pas commencer ou se terminer par un trait d'union (`-`)
  - Ils ne peuvent contenir ni traits de soulignement (_), ni lettres majuscules, ni points (`.`)
  - Il est impossible de les modifier après qu'ils aient été créés
- Les alias des points d'accès multi-régions sont générés par Amazon S3 et ne peuvent pas être modifiés ou réutilisés.
- Vous ne pouvez pas accéder à des données via un point d'accès multi-région en utilisant des points de terminaison de passerelle. En revanche, vous pouvez accéder aux données via un point d'accès multi-région en utilisant des points de terminaison d'interface. Pour l'utiliser AWS PrivateLink, vous devez créer des points de terminaison VPC. Pour plus d'informations, consultez Configurer un point d'accès multi-Régions pour utilisation avec AWS PrivateLink.
- Pour utiliser des points d'accès multirégionaux avec Amazon CloudFront, vous devez configurer le point d'accès multirégional en tant que type de `Custom Origin` distribution. Pour plus d'informations sur les différents types d'origine, consultez la section Utilisation de différentes origines avec CloudFront les distributions. Pour plus d'informations sur l'utilisation de points d'accès multirégionaux avec Amazon CloudFront, consultez la section Création d'une application active-active basée sur la proximité dans plusieurs régions sur le blog sur le stockage.AWS
- Exigences minimales relatives aux points d'accès multi-régions :
  - Transport Layer Security (TLS) v1.2
  - Prise en charge de Signature version 4 (SigV4A)

    Les points d'accès multi-régions prennent en charge Signature version 4A. Cette version de SigV4 permet de signer des requêtes pour plusieurs Régions AWS. Cette fonction est utile dans

les opérations d'API qui peuvent entraîner un accès aux données à partir d'une de plusieurs régions. Lorsque vous utilisez un AWS SDK, vous fournissez vos informations d'identification, et les demandes adressées aux points d'accès multirégionaux utiliseront la version 4A de Signature sans configuration supplémentaire. Assurez-vous de vérifier la compatibilité de votre kit AWS SDK avec l'algorithme SigV4A. Pour plus d'informations sur SIGv4a, consultez la section Signature des demandes AWS d'API dans le. Références générales AWS

> **Note**
>
> Pour utiliser SigV4A avec des informations d'identification de sécurité temporaires, par exemple, lorsque vous utilisez des rôles AWS Identity and Access Management (IAM), vous pouvez demander les informations d'identification temporaires à un point de terminaison régional (). AWS Security Token Service AWS STS Si vous demandez des informations d'identification temporaires au point de AWS STS terminaison global (`sts.amazonaws.com`), vous devez d'abord définir la compatibilité régionale des jetons de session pour que le point de terminaison global soit valide dans tous les cas Régions AWS. Pour plus d'informations, consultez la section Gestion AWS STS dans et Région AWS dans le guide de l'utilisateur IAM.

- Les points d'accès multi-régions ne prennent pas en charge les demandes anonymes.
- Limitations des points d'accès multi-régions :
  - IPv6 n'est pas pris en charge.
  - Les compartiments Amazon S3 sur Outposts ne sont pas pris en charge.
  - Les points d'accès multirégionaux prennent en charge les opérations de copie utilisant des points d'accès multirégionaux uniquement comme destination lors de l'utilisation de l'ARN du point d'accès multirégional.
  - La fonction d'opérations par lots S3 n'est pas prise en charge.
- Certains AWS SDK ne sont pas pris en charge. Pour vérifier quels AWS SDK sont pris en charge pour les points d'accès multirégionaux, consultez la section Compatibilité avec AWS les SDK.
- Les quotas de service pour les points d'accès multi-régions sont les suivants :
  - Il existe une limite maximale de 100 points d'accès multi-régions par compte.
  - Il existe une limite de 17 régions pour un seul point d'accès multi-régions.
- Une fois le point d'accès multi-régions créé, vous ne pouvez pas ajouter, modifier ou supprimer des compartiments de la configuration du point d'accès multi-régions. Pour modifier les compartiments, vous devrez supprimer tout le point d'accès multi-Régions et en créer un autre. Si un compartiment

multi-comptes de votre point d'accès multi-régions est supprimé, la seule façon de reconnecter ce compartiment est de le recréer, en utilisant le même nom et la même région dans ce compte.

- Les compartiments sous-jacents (dans le même compte) utilisés dans un point d'accès multi-régions peuvent être supprimés uniquement si ce point d'accès multi-régions est supprimé.
- Toutes les demandes du plan de contrôle de création ou de gestion de points d'accès multirégions doivent être routées vers la région `US West (Oregon)`. Pour les demandes de plan de données de points d'accès multirégions, il n'est pas nécessaire de spécifier les régions.
- Pour le plan de contrôle de basculement des points d'accès multirégions, les demandes doivent être routées vers l'une des cinq régions prises en charge suivantes :
  - `US East (N. Virginia)`
  - `US West (Oregon)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Europe (Ireland)`
- Votre point d'accès multirégional ne prend en charge que les compartiments suivants : Régions AWS
  - `US East (N. Virginia)`
  - `US East (Ohio)`
  - `US West (N. California)`
  - `US West (Oregon)`
  - `Asia Pacific (Mumbai)`
  - `Asia Pacific (Osaka)`
  - `Asia Pacific (Seoul)`
  - `Asia Pacific (Singapore)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Canada (Central)`
  - `Europe (Frankfurt)`
  - `Europe (Ireland)`
  - `Europe (London)`
  - `Europe (Paris)`

- `Europe (Stockholm)`
- `South America (São Paulo)`

Routage des demandes de points d'accès multi-Régions

Lorsque vous effectuez une demande via un point d'accès multi-régions, Amazon S3 détermine lequel des compartiments associés au point d'accès multi-régions est le plus proche de vous. Amazon S3 dirige ensuite la demande vers ce compartiment, quelle que soit la Région AWS dans laquelle il est situé.

Une fois que le point d'accès multi-régions a acheminé la demande vers le compartiment le plus proche, Amazon S3 traite la demande comme si vous l'aviez adressée directement à ce compartiment. Les points d'accès multi-régions ne connaissent pas le contenu des données d'un compartiment Amazon S3. Par conséquent, le compartiment qui reçoit la demande peut ne pas contenir les données demandées. Pour créer des jeux de données cohérents dans les compartiments Amazon S3 associés à un point d'accès multi-régions, vous pouvez configurer la réplication entre régions (CRR) S3. Ensuite, n'importe quel compartiment peut répondre à la demande avec succès.

Amazon S3 dirige les demandes de points d'accès multi-Régions conformément aux règles suivantes :

- Amazon S3 optimise l'exécution des requêtes en fonction de leur proximité. Il examine les compartiments pris en charge par le point d'accès multi-régions et transmet la requête au compartiment le plus proche.
- Si la demande indique une ressource existante (par exemple, `GetObject`), Amazon S3 peut ne pas tenir compte du nom de l'objet lorsqu'il exécutera la demande. Cela signifie que même si un objet existe dans un compartiment dans le point d'accès multi-régions, votre demande peut être acheminée vers un compartiment qui ne contient pas l'objet. Cela entraînera un message d'erreur 404 retourné au client.

  Pour éviter les erreurs 404, nous vous recommandons de configurer la réplication entre régions (CRR) S3 pour vos compartiments. La réplication permet de résoudre le problème potentiel lorsque l'objet que vous voulez est dans un compartiment dans le point d'accès multi-régions, mais qu'il ne se trouve pas dans le compartiment spécifique vers lequel votre demande a été acheminée. Pour obtenir des informations sur la configuration de réplication de base, consultez Configuration de la réplication à utiliser avec des points d'accès multi-régions.

Pour vous assurer que vos demandes sont traitées à l'aide des objets spécifiques que vous souhaitez, nous vous recommandons également d'activer la gestion des versions du compartiment et d'inclure des ID de version dans vos demandes. Cette approche permet de vous assurer que vous disposez de la version correcte de l'objet que vous recherchez. Les compartiments avec la gestion des versions activée vous permettent de récupérer des objets en cas de remplacement accidentel. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

- Si la demande vise à créer une ressource (par exemple, `PutObject` ou `CreateMultipartUpload`), Amazon S3 répond à la demande en utilisant le compartiment le plus proche. Prenons l'exemple d'une société de vidéo qui souhaite prendre en charge les chargements de vidéos de n'importe où dans le monde vers le compartiment le plus proche. Lorsqu'un utilisateur effectue une demande PUT au point d'accès multi-régions, l'objet est placé dans le compartiment le plus proche. Pour ensuite permettre à des utilisateurs du monde entier de télécharger cette vidéo avec la latence la plus faible, vous pouvez utiliser le CRR avec une réplication bidirectionnelle. L'utilisation du CRR avec une réplication bidirectionnelle permet de synchroniser le contenu de tous les compartiments associés au point d'accès multi-régions. Pour plus d'informations sur la réplication avec des points d'accès multi-régions, consultez Configuration de la réplication à utiliser avec des points d'accès multi-régions.

Contrôles de basculement des points d'accès multi-régions Amazon S3

Grâce aux contrôles de basculement du point d'accès multi-régions d'Amazon S3, vous pouvez maintenir la continuité des activités pendant les perturbations du trafic régional, tout en offrant à vos applications une architecture multi-régions pour répondre aux besoins de conformité et de redondance. Si votre trafic régional est perturbé, vous pouvez utiliser les contrôles de basculement du point d'accès multi-régions pour sélectionner la Régions AWS derrière un point d'accès multi-régions Amazon S3 qui traitera les demandes d'accès aux données et de stockage.

Pour prendre en charge le basculement, vous pouvez configurer votre point d'accès multi-régions dans une configuration active-passive, le trafic circulant vers la région active dans des conditions normales, et une région passive de secours pour le basculement.

Par exemple, pour effectuer un basculement vers une Région AWS de votre choix, vous déplacez le trafic de votre région principale (active) vers votre région secondaire (passive). Dans une configuration active-passive comme celle-ci, un compartiment est actif et accepte le trafic, tandis que l'autre est passif et n'accepte pas le trafic. Le compartiment passif est utilisé pour la reprise après

sinistre. Lorsque vous lancez le basculement, tout le trafic (tel que les requêtes GET et PUT) est dirigé vers le compartiment actif (dans une région) et éloigné du compartiment passif (dans une autre région).

Si vous avez activé la réplication entre régions (CRR) de S3 avec des règles de réplication bidirectionnelle, vous pouvez garder vos compartiments synchronisés pendant un basculement. En outre, si vous avez activé la fonction CRR dans une configuration active-active, les points d'accès multi-régions d'Amazon S3 peuvent également récupérer des données à partir de l'emplacement du compartiment le plus proche, ce qui améliore les performances des applications.

Prise en charge d'Région AWS

Grâce aux contrôles de basculement des points d'accès multi-régions d'Amazon S3, vos compartiments S3 peuvent se trouver dans l'une des 17 régions où les points d'accès multi-régions sont pris en charge. Vous pouvez initier un basculement sur deux régions à la fois.

> **Note**
>
> Bien que le basculement ne soit initié qu'entre deux régions à la fois, vous pouvez mettre à jour séparément les statuts de routage pour plusieurs régions en même temps dans votre point d'accès multi-régions.

Les rubriques suivantes expliquent comment utiliser et gérer les contrôles de basculement du point d'accès multi-régions Amazon S3.

Rubriques

- États d'acheminement des points d'accès multi-régions Amazon S3
- Utilisation des contrôles de basculement du point d'accès multi-régions d'Amazon S3
- Erreurs de contrôle du basculement du point d'accès multi-régions Amazon S3

États d'acheminement des points d'accès multi-régions Amazon S3

Votre configuration de basculement des points d'accès multi-régions Amazon S3 détermine le statut de routage de la Régions AWS utilisée avec le point d'accès multi-régions. Vous pouvez configurer votre point d'accès multi-régions Amazon S3 pour qu'il soit dans un état actif-actif ou actif-passif.

- Active-active : dans une configuration active-active, toutes les requêtes sont automatiquement envoyées vers la Région AWS la plus proche de votre point d'accès multi-régions. Une fois que le point d'accès multi-régions a été configuré pour être dans un état actif-actif, toutes les régions peuvent recevoir du trafic. Si une interruption du trafic se produit dans une configuration active-active, le trafic réseau sera automatiquement redirigé vers l'une des régions actives.
- Active-passive : dans une configuration active-passive, les régions actives de votre point d'accès multi-régions reçoivent du trafic et les régions passives n'en reçoivent pas. Si vous avez l'intention d'utiliser les contrôles de basculement S3 pour initier le basculement en cas de sinistre, configurez vos points d'accès multi-régions dans une configuration active-passive pendant les tests et la planification de la reprise après sinistre.

Utilisation des contrôles de basculement du point d'accès multi-régions d'Amazon S3

Cette section explique comment gérer et utiliser les contrôles de basculement de vos points d'accès multi-régions Amazon S3 à l'aide de la fonction AWS Management Console.

Il existe deux commandes de basculement dans la section Failover configuration (Configuration du basculement) de la page de détails de votre point d'accès multi-régions, dans la AWS Management Console : Edit routing status (Modifier l'état du routage) et Failover (Basculement). Vous pouvez utiliser ces contrôles comme suit :

- Edit routing status (Modifier le statut de routage) : vous pouvez modifier manuellement le statut de routage d'un maximum de 17 Régions AWS en une seule demande pour votre point d'accès multi-régions en sélectionnant Edit routing status (Modifier le statut de routage). Vous pouvez utiliser la fonction Edit routing status (Modifier le statut de routage) pour les raisons suivantes :
  - Pour définir ou modifier les statuts d'acheminement d'une ou plusieurs régions dans votre point d'accès multi-régions.
  - Pour créer une configuration de basculement pour votre point d'accès multi-régions en configurant deux régions dans un état actif-passif.
  - Pour basculer manuellement sur vos régions
  - Pour basculer manuellement le trafic entre les régions
- Failover (Basculement): lorsque vous lancez le basculement en sélectionnant Failover (Basculement), vous ne faites que mettre à jour les statuts d'acheminement de deux régions qui sont déjà configurées pour être dans un état actif-passif. Lors d'un basculement que vous avez déclenché en sélectionnant Failover (Basculement), les statuts d'acheminement entre les deux régions sont automatiquement permutés.

Modification du statut de routage des régions dans votre point d'accès multi-régions.

Vous pouvez mettre à jour manuellement les statuts d'acheminement de jusqu'à 17 Régions AWS en une seule demande pour votre point d'accès multi-régions en choisissant Edit routing status (Modifier le statut de routage) dans la section Failover configuration (Configuration du basculement) sur la page de détails de votre point d'accès multi-régions. Cependant, lorsque vous lancez le basculement en sélectionnant Failover (Basculement), vous ne faites que mettre à jour les statuts de routage de deux régions qui sont déjà configurées pour être dans un état actif-passif. Lors d'un basculement que vous avez déclenché en sélectionnant Failover (Basculement), les statuts d'acheminement entre les deux régions sont automatiquement permutés.

Vous pouvez utiliser l'option Edit routing status (Modifier le statut de routage) comme décrit dans la procédure suivante, aux fins suivantes :

- Pour définir ou modifier les statuts d'acheminement d'une ou plusieurs régions dans votre point d'accès multi-régions.
- Pour créer une configuration de basculement pour votre point d'accès multi-régions en configurant deux régions dans un état actif-passif.
- Pour basculer manuellement sur vos régions
- Pour basculer manuellement le trafic entre les régions

Utilisation de la console S3

Pour mettre à jour le statut de routage des régions dans votre point d'accès multi-régions

1. Connectez-vous à AWS Management Console.
2. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
3. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
4. Choisissez le point d'accès multi-régions que vous souhaitez mettre à jour.
5. Choisissez l'onglet Replication and failover (Réplication et basculement).
6. Sélectionnez une ou plusieurs régions dont vous voulez modifier le statut de routage.

> **Note**
>
> Pour initier le basculement, au moins une Région AWS doit être désignée comme Active et une région doit être désignée comme Passive dans votre point d'accès multi-régions.

7. Choisissez Edit routing status (Modifier le statut du routage).
8. Dans la boîte de dialogue qui s'affiche, sélectionnez Active ou Passive pour le Routing status (Statut de routage) de chaque région.

   Un état actif permet de router le trafic vers la région. Un état passif empêche tout trafic d'être dirigé vers la région.

   Si vous créez une configuration de basculement pour votre point d'accès multi-régions ou si vous initiez un basculement, au moins une Région AWS doit être désignée comme Active et une région doit être désignée comme Passive dans votre point d'accès multi-régions.
9. Sélectionnez Save routing status (Enregistrer le statut de routage). Il faut environ deux minutes pour que le trafic soit redirigé.

Après avoir soumis le statut de routage des Régions AWS pour votre point d'accès multi-régions, vous pouvez vérifier les changements de votre statut de routage. Pour vérifier ces changements, rendez-vous sur Amazon CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/ pour surveiller le déplacement de votre trafic de requêtes de données Amazon S3 (par exemple, `GET` et `PUT`) entre les régions actives et passives. Les connexions existantes ne seront pas interrompues pendant le basculement. Les connexions existantes se poursuivront jusqu'à ce qu'elles atteignent un statut de réussite ou d'échec.

Utilisation de AWS CLI

> **Note**
>
> Vous pouvez exécuter des commandes de routage AWS CLI de points d'accès multi-régions dans n'importe laquelle de ces cinq régions :
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`
> - `us-west-2`

- eu-west-1

L'exemple de commande suivant met à jour la configuration des itinéraires de votre point d'accès multi-régions actuel. Pour mettre à jour l'état actif ou passif d'un compartiment, définissez la valeur `TrafficDialPercentage` sur `100` pour actif et sur `0` pour passif. Dans cet exemple, la valeur *DOC-EXAMPLE-BUCKET-1* est réglée sur actif, et *DOC-EXAMPLE-BUCKET-2* est réglée sur passif. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
--route-updates Bucket=DOC-EXAMPLE-BUCKET-1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET-2,TrafficDialPercentage=0
```

L'exemple de commande suivant permet d'obtenir votre configuration de routage de point d'accès multi-régions mise à jour. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap MultiRegionAccessPoint_ARN
```

Initialisation du basculement

Lorsque vous lancez le basculement en sélectionnant Failover (Basculement) dans la section Failover configuration (Configuration du basculement) sur la page de détails de votre point d'accès multi-régions, le trafic des demandes Amazon S3 est automatiquement transféré vers une autre Région AWS. Le processus de basculement s'achève dans les deux minutes.

Vous pouvez déclencher un basculement sur deux Régions AWS à la fois (parmi les 17 régions où les points d'accès multi-régions sont pris en charge). Les événements de basculement sont ensuite consignés dans AWS CloudTrail. Une fois le basculement terminé, vous pouvez surveiller le trafic Amazon S3 et toutes les mises à jour de routage du trafic vers la nouvelle région active dans Amazon CloudWatch.

> **Important**
>
> Pour que toutes les métadonnées et tous les objets soient synchronisés entre les compartiments pendant la réplication des données, nous vous recommandons de créer des règles de réplication bidirectionnelle et d'activer la synchronisation des modifications des réplicas avant de configurer vos contrôles de basculement.
> Les règles de réplication bidirectionnelle permettent de s'assurer que lorsque des données sont écrites dans le compartiment Amazon S3 vers lequel le trafic bascule, ces données sont ensuite répliquées vers le compartiment source. La synchronisation des modifications des réplicas permet de s'assurer que les métadonnées des objets sont également synchronisées entre les compartiments pendant la réplication bidirectionnelle.
> Pour plus d'informations sur la configuration de la réplication pour prendre en charge le basculement, consultez the section called “Réplication de compartiment”.

Pour initier le basculement entre les compartiments répliqués

1. Connectez-vous à AWS Management Console.
2. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
3. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
4. Choisissez le point d'accès multi-régions que vous voulez utiliser pour initier le basculement.
5. Choisissez l'onglet Replication and failover (Réplication et basculement).
6. Descendez jusqu'à la section Failover configuration (Configuration du basculement) et sélectionnez deux Régions AWS.

   > **Note**
   >
   > Pour initier le basculement, au moins une Région AWS doit être désignée comme Active et une région doit être désignée comme Passive dans votre point d'accès multi-régions. Un état actif permet de diriger le trafic vers une région. Un état passif empêche tout trafic d'être dirigé vers la région.

7. Choisissez Failover (Basculement).
8. Dans la boîte de dialogue, sélectionnez Failover (Basculement) à nouveau pour lancer le processus de basculement. Au cours de ce processus, les statuts de routage des deux régions

sont automatiquement permutés. Tout nouveau trafic est dirigé vers la région qui devient active, et le trafic cesse d'être dirigé vers la région qui devient passive. Il faut environ deux minutes pour que le trafic soit redirigé.

Après avoir lancé le processus de basculement, vous pouvez contrôler vos modifications de trafic. Pour vérifier ces changements, rendez-vous sur Amazon CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/ pour surveiller le déplacement de votre trafic de requêtes de données Amazon S3 (par exemple, `GET` et `PUT`) entre les régions actives et passives. Les connexions existantes ne seront pas interrompues pendant le basculement. Les connexions existantes se poursuivront jusqu'à ce qu'elles atteignent un statut de réussite ou d'échec.

Visualisation des contrôles de routage de votre point d'accès multi-régions Amazon S3

Utilisation de la console S3

Pour afficher les commandes de routage de votre point d'accès Multi-Régions Amazon S3

1. Connectez-vous à AWS Management Console.
2. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
3. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
4. Choisissez le point d'accès multi-régions que vous voulez examiner.
5. Choisissez l'onglet Replication and failover (Réplication et basculement). Cette page affiche les détails et la synthèse de la configuration du routage pour votre point d'accès multi-régions, les règles de réplication associées et les métriques de réplication. Vous pouvez voir le statut de routage de vos régions dans la section Failover configuration (Configuration du basculement).

Utilisation de AWS CLI

L'exemple de commande suivant de la AWS CLI permet d'obtenir la configuration actuelle de l'itinéraire du point d'accès multi-régions pour la région spécifiée. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
```

```
--mrap MultiRegionAccessPoint_ARN
```

> **Note**
>
> Cette commande ne peut être exécutée que sur les cinq régions suivantes :
>
> - ap-southeast-2
> - ap-northeast-1
> - us-east-1
> - us-west-2
> - eu-west-1

Erreurs de contrôle du basculement du point d'accès multi-régions Amazon S3

Lorsque vous mettez à jour la configuration du basculement pour votre point d'accès multi-régions, vous pouvez rencontrer l'une des erreurs suivantes :

- HTTP 400 Demande erronée : ce problème peut survenir si vous entrez un ARN de point d'accès multi-régions non valide lors de la mise à jour de votre configuration de basculement. Vous pouvez confirmer votre ARN de point d'accès multi-régions en consultant votre politique de point d'accès multi-régions. Pour réviser ou mettre à jour votre politique de point d'accès multi-régions, consultez Editing the Multi-Region Access Point policy (Modification de la politique de point d'accès multi-régions). Cette erreur peut également se produire si vous utilisez une chaîne vide ou une chaîne aléatoire lors de la mise à jour de vos contrôles de basculement du point d'accès multi-régions Amazon S3. Veillez à utiliser le format ARN du point d'accès multi-régions :

  `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`
- HTTP 503 Slow Down (HTTP 503 Ralentissement) : cette erreur se produit si vous envoyez trop de requêtes dans un court laps de temps. Les requêtes rejetées donneront lieu à une erreur.
- HTTP 409 Conflict (HTTP 409 Conflit) : cette erreur survient lorsque deux ou plusieurs requêtes simultanées de mise à jour de la configuration des itinéraires visent un seul point d'accès multi-régions. La première requête aboutit, mais toutes les autres échouent avec une erreur.
- HTTP 405 Method Not Allowed (Méthode HTTP 405 non autorisée) : cette erreur se produit lorsque vous avez sélectionné un point d'accès multi-régions avec une seule Région AWS lors du lancement du basculement. Vous devez sélectionner deux régions avant de pouvoir lancer le basculement. Sinon, une erreur est renvoyée.

Configuration de la réplication à utiliser avec des points d'accès multi-régions

Lorsque vous adressez une demande à un point de terminaison d'accès multi-régions, Amazon S3 achemine automatiquement la demande vers le compartiment le plus proche de vous. Amazon S3 ne tient pas compte du contenu de la demande lorsqu'il prend cette décision. Si vous envoyez une demande pour `GET` un objet, votre demande peut être acheminée vers un compartiment qui n'a pas de copie de cet objet. Si c'est le cas, vous recevez un code d'erreur HTTP 404 (Not Found) [HTTP 404 (Introuvable)]. Pour plus d'informations sur l'acheminement d'une demande d'un point d'accès multi-régions, consultez the section called “Routage des demandes”.

Si vous voulez que le point d'accès multi-régions puisse récupérer l'objet quel que soit le compartiment qui reçoit la requête, vous devez configurer la réplication entre régions (CRR) d'Amazon S3.

Par exemple, prenons un point d'accès multi-régions avec trois compartiments :

- Un compartiment nommé `my-bucket-usw2` dans la région `us-west-2` qui contient un objet `my-image.jpg`
- Un compartiment nommé `my-bucket-aps1` dans la région `ap-south-1` qui contient un objet `my-image.jpg`
- Un compartiment nommé `my-bucket-euc1` dans la région `eu-central-1` qui ne contient pas l'objet `my-image.jpg`

Dans cette situation, si vous faites une demande `GetObject`pour l'objet `my-image.jpg`, la réussite de cette demande dépend du compartiment qui reçoit votre demande. Étant donné qu'Amazon S3 ne tient pas compte du contenu de la requête, il se peut que votre requête `GetObject` soit acheminée vers le compartiment `my-bucket-euc1` le plus proche. Même si votre objet se trouve dans un compartiment du point d'accès multi-régions, vous obtiendrez une erreur HTTP 404 Not Found [HTTP 404 (Introuvable)], car le compartiment individuel qui a reçu votre requête ne possède pas l'objet.

L'activation de la réplication entre régions (CRR) permet d'éviter ce résultat. Avec les règles de réplication appropriées, l'objet `my-image.jpg` est copié dans le compartiment `my-bucket-euc1`. Par conséquent, si Amazon S3 achemine votre requête vers ce compartiment, vous pouvez maintenant récupérer l'objet.

La réplication fonctionne normalement avec des compartiments qui sont affectés à un point d'accès multi-Régions. Amazon S3 n'effectue pas de traitement spécial de réplication avec les compartiments qui se trouvent dans des points d'accès multi-régions. Pour en savoir plus sur la configuration de la réplication dans vos compartiments, consultez Configuration de la réplication en direct.

Recommandations pour l'utilisation de la réplication avec des points d'accès multi-régions

Pour obtenir les meilleures performances de réplication lorsque vous travaillez avec des points d'accès multi-régions, nous vous recommandons ce qui suit :

- Configurez le contrôle du temps de réplication S3 (S3 RTC). Pour répliquer vos données sur différentes régions dans un délai prévisible, vous pouvez utiliser S3 RTC. Le contrôle du délai de réplication S3 permet de répliquer 99,99 % des nouveaux objets stockés dans Simple Storage Service (Amazon S3) dans les 15 minutes (conformément à un contrat de niveau de service (SLA)). Pour de plus amples informations, veuillez consulter the section called “Utiliser le contrôle du délai de réplication S3”. Des frais supplémentaires sont facturés pour l'utilisation de S3 RTC. Pour plus d'informations, consultez Tarification Amazon S3.
- Utilisez la réplication bidirectionnelle pour maintenir les compartiments synchronisés lorsque les compartiments sont mis à jour par le point d'accès multi-régions. Pour de plus amples informations, veuillez consulter the section called “Création de règles de réplication bidirectionnelle pour votre point d'accès multi-régions”.
- Créez des points d'accès multi-régions intercompte pour répliquer des données dans des compartiments dans différents Comptes AWS. Cette approche permet une séparation au niveau des comptes, de sorte que les données peuvent être consultées et répliquées sur différents comptes dans différentes régions autres que le compartiment source. La configuration de points d'accès multi-régions intercompte n'entraîne aucun coût supplémentaire. Si vous êtes propriétaire d'un compartiment mais que vous ne possédez pas le point d'accès multi-régions, vous ne payez que les frais de transfert de données et de demande. Les propriétaires de points d'accès multi-régions paient pour l'acheminement des données et les coûts d'accélération d'Internet. Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
- Activez la synchronisation des modifications des réplicas pour chaque règle de réplication afin de synchroniser également les modifications des métadonnées de vos objets. Pour plus d'informations, consultez Enabling replica modification sync (Activation de la synchronisation des modifications de réplicas).
- Activez les métriques Amazon CloudWatch pour surveiller les événements de réplication. Des frais supplémentaires peuvent s'appliquer lorsque vous utilisez les métriques de CloudWatch. Pour de plus amples informations, consultez Tarification Amazon CloudWatch.

Rubriques

Création de règles de réplication unidirectionnelle pour votre point d'accès multi-régions

Les règles de réplication permettent de copier automatiquement et de manière asynchrone les objets d'un compartiment à l'autre. Une règle de réplication unidirectionnelle permet de garantir que les données sont entièrement répliquées depuis un compartiment source d'une Région AWS vers un compartiment de destination dans une autre région. Lorsque la réplication unidirectionnelle est configurée, le système crée une règle de réplication du compartiment source (DOC-EXAMPLE-BUCKET-1) vers le compartiment de destination (DOC-EXAMPLE-BUCKET-2). Comme toutes les règles de réplication, vous pouvez appliquer la règle de réplication unidirectionnelle à l'ensemble du compartiment Amazon S3 ou à un sous-ensemble d'objets filtrés par un préfixe ou des étiquettes d'objet.

> **Important**
>
> Nous vous recommandons d'utiliser la réplication unidirectionnelle si vos utilisateurs ne consomment que des objets de vos compartiments de destination. Si vos utilisateurs doivent télécharger ou modifier les objets de vos compartiments de destination, utilisez la réplication bidirectionnelle pour synchroniser tous vos compartiments. Nous recommandons également la réplication bidirectionnelle si vous prévoyez d'insérer le point d'accès multi-régions à des fins de basculement. Pour configurer la réplication bidirectionnelle, consultez the section called “Création de règles de réplication bidirectionnelle pour votre point d'accès multi-régions”.

Pour créer une règle de réplication unidirectionnelle pour votre point d'accès multi-régions

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).

3. Choisissez le nom de votre point d'accès multi-régions.
4. Choisissez l'onglet Replication and failover (Réplication et basculement).
5. Faites défiler l'écran jusqu'à la section Replication rules (Règles de réplication), puis sélectionnez Create replication rules (Créer des règles de réplication). Assurez-vous que vous disposez des autorisations suffisantes pour créer la règle de réplication. Dans le cas contraire, la gestion des versions sera désactivée.

> **Note**
>
> Vous pouvez créer des règles de réplication uniquement pour les compartiments dans votre compte. Pour créer des règles de réplication pour des compartiments externes, les propriétaires des compartiments doivent créer les règles de réplication pour ces compartiments.

6. Sur la page Créer des règles de réplication, choisissez le modèle Répliquer les objets d'un ou de plusieurs compartiments sources vers un ou plusieurs compartiments de destination.

> **Important**
>
> Lorsque vous créez des règles de réplication à l'aide de ce modèle, elles remplacent toutes les règles de réplication existantes qui sont déjà affectées au compartiment. Pour ajouter ou modifier une règle de réplication existante au lieu de la remplacer, accédez à l'onglet Management (Gestion) de chaque compartiment dans la console, puis modifiez la règle dans la section Replication rules (Règles de réplication). Vous pouvez également ajouter ou modifier des règles de réplication existantes en utilisant la AWS CLI, les kits SDK ou l'API REST. Pour de plus amples informations, veuillez consulter Configuration de réplication.

7. Dans la section Source et destination, sous Compartiments source, sélectionnez un ou plusieurs compartiments à partir desquels vous souhaitez répliquer les objets. Tous les compartiments (source et destination) choisis pour la réplication doivent avoir l'option de gestion des versions S3 activée, et chaque compartiment doit résider dans une Région AWS différente. Pour plus d'informations sur la gestion des versions S3, consultez la section Using versioning in Amazon S3 buckets (Utilisation de la la gestion des versions dans les compartiments Amazon S3).

Sous Compartiments de destination, sélectionnez un ou plusieurs compartiments à partir desquels vous souhaitez répliquer les objets.

> **Note**
>
> Assurez-vous de disposer des autorisations de lecture et de réplication requises pour établir la réplication, sinon vous risquez de rencontrer des erreurs. Pour plus d'informations, consultez Création d'un rôle IAM.

8. Dans la section Replication rule configuration (Configuration de la règle de réplication), choisissez si la règle de réplication sera Enabled (Activée) ou Disabled (Désactivée) lors de sa création.

> **Note**
>
> Vous ne pouvez pas saisir de nom dans la case Replication rule name (Nom de la règle de réplication). Les noms des règles de réplication sont générés en fonction de votre configuration lorsque vous créez la règle de réplication.

9. Dans la section Scope (Étendue), choisissez l'étendue appropriée pour votre réplication.

   - Pour répliquer l'ensemble du compartiment, choisissez Apply to all objects in the bucket (Appliquer à tous les objets du compartiment).
   - Pour répliquer un sous-ensemble des objets du compartiment, sélectionnez Limit the scope of this rule using one or more filters (Limiter l'étendue de cette règle à l'aide d'un ou plusieurs filtres).

     Vous pouvez filtrer vos objets en utilisant un préfixe, des étiquettes d'objets ou une combinaison des deux.

     - Pour limiter la réplication à tous les objets dont le nom commence par la même chaîne (par exemple `pictures`), saisissez un préfixe dans la case Prefix (Préfixe).

       Si vous saisissez un préfixe correspondant à un nom de dossier, vous devez insérer un caractère `/` à la fin pour indiquer son niveau de hiérarchie (par exemple, `pictures/`). Pour plus d'informations sur les préfixes, consultez Organisation des objets à l'aide de préfixes.
     - Pour répliquer tous les objets avec une ou plusieurs étiquettes d'objet, sélectionnez Add tag (Ajouter une étiquette) et saisissez la paire clé-valeur dans les zones. Pour ajouter une

autre étiquette, répétez la procédure. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.

10. Faites défiler vers le bas jusqu'à la section Additional replication options (Options de réplication supplémentaires), puis sélectionnez les options de réplication que vous souhaitez appliquer.

> **Note**
>
> Nous vous recommandons d'appliquer les options suivantes :
>
> - Replication time control (RTC) [Contrôle du temps de réplication (RTC)] : pour répliquer vos données sur différentes régions dans un délai prévisible, vous pouvez utiliser le contrôle du temps de réplication S3 (S3 RTC). Le contrôle du délai de réplication S3 permet de répliquer 99,99 % des nouveaux objets stockés dans Simple Storage Service (Amazon S3) dans les 15 minutes (conformément à un contrat de niveau de service (SLA)). Pour de plus amples informations, veuillez consulter the section called “Utiliser le contrôle du délai de réplication S3”.
> - Replication metrics and notifications (Métriques et notifications de réplication) : activez les métriques Amazon CloudWatch pour surveiller les événements de réplication.
> - Réplication des marqueurs de suppression : les marqueurs de suppression créés par les opérations de suppression de S3 seront répliqués. Les marqueurs de suppression créés par les règles de cycle de vie ne sont pas répliqués. Pour plus d'informations, consultez Répliquer des marqueurs de suppression entre les compartiments.
>
> Des frais supplémentaires sont facturés pour les métriques et les notifications de réplication S3 RTC et CloudWatch. Pour plus d'informations, consultez Amazon S3 Pricing (Tarification d'Amazon S3) et Amazon CloudWatch pricing (Tarification d'Amazon CloudWatch).

11. Si vous écrivez une nouvelle règle de réplication qui remplace une règle existante, sélectionnez I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Je reconnais qu'en choisissant l'option Créer des règles de réplication, ces règles de réplication existantes seront écrasées).
12. Choisissez Créer des règles de réplication pour créer et enregistrer votre nouvelle règle de réplication unidirectionnelle.

Création de règles de réplication bidirectionnelle pour votre point d'accès multi-régions

Les règles de réplication permettent de copier automatiquement et de manière asynchrone les objets d'un compartiment à l'autre. Une règle de réplication bidirectionnelle garantit que les données sont entièrement synchronisées entre deux ou plusieurs compartiments dans différentes Régions AWS. Lorsque la réplication bidirectionnelle est configurée, le système crée une règle de réplication du compartiment source (DOC-EXAMPLE-BUCKET-1) vers le compartiment contenant les réplicas (DOC-EXAMPLE-BUCKET-2). Ensuite, une deuxième règle de réplication est créée depuis le compartiment contenant les réplicas (DOC-EXAMPLE-BUCKET-2) vers le compartiment source (DOC-EXAMPLE-BUCKET-1).

Comme toutes les règles de réplication, vous pouvez appliquer la règle de réplication bidirectionnelle à l'ensemble du compartiment Amazon S3 ou à un sous-ensemble d'objets filtrés par un préfixe ou des étiquettes d'objet. Vous pouvez également synchroniser les modifications des métadonnées de vos objets en activant la synchronisation des modifications des réplicas pour chaque règle de réplication. Vous pouvez activer la synchronisation des modifications de réplicas via la console Amazon S3, l'AWS CLI, les kits AWS SDK, l'API REST Amazon S3 ou AWS CloudFormation.

Pour surveiller la progression de la réplication des objets et des métadonnées d'objets dans Amazon CloudWatch, activez les métriques et les notifications de réplication S3. Pour plus d'informations, consultez Monitoring progress with replication metrics and Amazon S3 event notifications (Suivi de la progression avec les métriques de réplication et les notifications d'événements Amazon S3).

Pour créer une règle de réplication bidirectionnelle pour votre point d'accès multi-régions

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).
3. Choisissez le nom du point d'accès multi-régions que vous souhaitez mettre à jour.
4. Choisissez l'onglet Replication and failover (Réplication et basculement).
5. Faites défiler l'écran jusqu'à la section Replication rules (Règles de réplication), puis sélectionnez Create replication rules (Créer des règles de réplication).
6. Sur la page Create replication rules (Créer des règles de réplication), choisissez le modèle Replicate objects among all specified buckets (Répliquer les objets parmi tous les compartiments spécifiés). Le modèle Replicate objects among all specified buckets (Répliquer les objets parmi

tous les compartiments spécifiés) configure une réplication bidirectionnelle (avec des capacités de basculement) pour vos compartiments.

> **Important**
>
> Lorsque vous créez des règles de réplication à l'aide de ce modèle, elles remplacent toutes les règles de réplication existantes qui sont déjà affectées au compartiment. Pour ajouter ou modifier une règle de réplication existante au lieu de la remplacer, accédez à l'onglet Management (Gestion) de chaque compartiment dans la console, puis modifiez la règle dans la section Replication rules (Règles de réplication). Vous pouvez également ajouter ou modifier des règles de réplication existantes en utilisant l'AWS CLI, les kits SDK AWS ou l'API REST Amazon S3. Pour de plus amples informations, veuillez consulter Configuration de réplication.

7. Dans la section Buckets (Compartiments), sélectionnez au moins deux compartiments à partir desquels vous souhaitez répliquer des objets. Tous les compartiments choisis pour la réplication doivent avoir l'option gestion des versions S3 activée, et chaque compartiment doit résider dans une Région AWS différente. Pour plus d'informations sur la gestion des versions S3, consultez la section Using versioning in Amazon S3 buckets (Utilisation de la la gestion des versions dans les compartiments Amazon S3).

> **Note**
>
> Assurez-vous de disposer des autorisations de lecture et de réplication requises pour établir la réplication, sinon vous risquez de rencontrer des erreurs. Pour plus d'informations, consultez Création d'un rôle IAM.

8. Dans la section Replication rule configuration (Configuration de la règle de réplication), choisissez si la règle de réplication sera Enabled (Activée) ou Disabled (Désactivée) lors de sa création.

> **Note**
>
> Vous ne pouvez pas saisir de nom dans la case Replication rule name (Nom de la règle de réplication). Les noms des règles de réplication sont générés en fonction de votre configuration lorsque vous créez la règle de réplication.

9. Dans la section Scope (Étendue), choisissez l'étendue appropriée pour votre réplication.

- Pour répliquer l'ensemble du compartiment, choisissez Apply to all objects in the bucket (Appliquer à tous les objets du compartiment).
- Pour répliquer un sous-ensemble des objets du compartiment, sélectionnez Limit the scope of this rule using one or more filters (Limiter l'étendue de cette règle à l'aide d'un ou plusieurs filtres).

  Vous pouvez filtrer vos objets en utilisant un préfixe, des étiquettes d'objets ou une combinaison des deux.

  - Pour limiter la réplication à tous les objets dont le nom commence par la même chaîne (par exemple `pictures`), saisissez un préfixe dans la case Prefix (Préfixe).

    Si vous entrez un préfixe correspondant à un nom de dossier, vous devez insérer le caractère `/` (barre oblique) en tant que dernier caractère (par exemple, `pictures/`).
  - Pour répliquer tous les objets avec une ou plusieurs étiquettes d'objet, sélectionnez Add tag (Ajouter une étiquette) et saisissez la paire clé-valeur dans les zones. Pour ajouter une autre étiquette, répétez la procédure. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.

10. Faites défiler vers le bas jusqu'à la section Additional replication options (Options de réplication supplémentaires), puis sélectionnez les options de réplication que vous souhaitez appliquer.

> **Note**
>
> Nous vous recommandons d'appliquer les options suivantes, en particulier si vous avez l'intention de configurer votre point d'accès multi-régions pour prendre en charge le basculement :
>
> - Replication time control (RTC) [Contrôle du temps de réplication (RTC)] : pour répliquer vos données sur différentes régions dans un délai prévisible, vous pouvez utiliser le contrôle du temps de réplication S3 (S3 RTC). Le contrôle du délai de réplication S3 permet de répliquer 99,99 % des nouveaux objets stockés dans Simple Storage Service (Amazon S3) dans les 15 minutes (conformément à un contrat de niveau de service (SLA)). Pour de plus amples informations, veuillez consulter the section called “Utiliser le contrôle du délai de réplication S3”.
> - Replication metrics and notifications (Métriques et notifications de réplication) : activez les métriques Amazon CloudWatch pour surveiller les événements de réplication.

- Réplication des marqueurs de suppression : les marqueurs de suppression créés par les opérations de suppression de S3 seront répliqués. Les marqueurs de suppression créés par les règles de cycle de vie ne sont pas répliqués. Pour plus d'informations, consultez Répliquer des marqueurs de suppression entre les compartiments.
- Replica modification sync (Synchronisation des modifications de réplicas) : activez la synchronisation des modifications des réplicas pour chaque règle de réplication afin de synchroniser également les modifications des métadonnées de vos objets. Pour plus d'informations, consultez Enabling replica modification sync (Activation de la synchronisation des modifications de réplicas).

Des frais supplémentaires sont facturés pour les métriques et les notifications de réplication S3 RTC et CloudWatch. Pour plus d'informations, consultez Amazon S3 Pricing (Tarification d'Amazon S3) et Amazon CloudWatch pricing (Tarification d'Amazon CloudWatch).

11. Si vous écrivez une nouvelle règle de réplication qui remplace une règle existante, sélectionnez I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Je reconnais qu'en choisissant l'option Créer des règles de réplication, ces règles de réplication existantes seront écrasées).
12. Choisissez Create replication rules (Créer des règles de réplication) pour créer et enregistrer vos nouvelles règles de réplication bidirectionnelle.

Affichage des règles de réplication pour votre point d'accès multi-régions

Avec les points d'accès multi-régions, vous pouvez configurer des règles de réplication unidirectionnelles ou des règles de réplication bidirectionnelles. Pour plus d'informations sur la gestion de vos règles de réplication, consultez Gérer les règles de réplication à l'aide de la console Amazon S3.

Pour afficher des règles de réplication pour votre point d'accès multi-régions

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Multi-Region Access Points (Points d'accès multi-régions).

3. Choisissez le nom de votre point d'accès multi-régions.
4. Choisissez l'onglet Replication and failover (Réplication et basculement).
5. Faites défiler jusqu'à la section Règles de réplication. Cette section répertorie toutes les règles de réplication créées pour votre point d'accès multi-régions.

> **Note**
>
> Si vous avez ajouté un compartiment provenant d'un autre compte à ce point d'accès multi-régions, vous devez disposer de l'autorisation `s3:GetBucketReplication` du propriétaire du compartiment pour consulter les règles de réplication de ce compartiment.

Utilisation de points d'accès multi-régions avec des opérations d'API prises en charge

Amazon S3 fournit un ensemble d'opérations permettant de gérer les points d'accès multi-régions. Amazon S3 traite certaines de ces opérations de manière synchrone et d'autres de manière asynchrone. Lorsque vous appelez une opération asynchrone, Amazon S3 autorise d'abord l'opération demandée de manière synchrone. Si l'autorisation réussit, Amazon S3 renvoie un jeton que vous pourrez utiliser pour suivre la progression et les résultats de l'opération demandée.

> **Note**
>
> Les demandes effectuées via la console Amazon S3 sont toujours synchrones. La console attend que la demande soit terminée avant de vous autoriser à envoyer une autre demande.

Vous pouvez consulter l'état actuel et les résultats des opérations asynchrones à l'aide de la console, ou vous pouvez les utiliser `DescribeMultiRegionAccessPointOperation` dans les AWS CLI AWS SDK ou l'API REST. Amazon S3 fournit un jeton de suivi dans la réponse à une opération asynchrone. Vous devrez inclure ce jeton de suivi en tant qu'argument de `DescribeMultiRegionAccessPointOperation`. Quand vous incluez le jeton de suivi, Amazon S3 renvoie ensuite le statut actuel et les résultats de l'opération spécifiée, notamment les éventuelles erreurs ou informations pertinentes sur les ressources. Amazon S3 effectue les opérations `DescribeMultiRegionAccessPointOperation` de manière synchrone.

Toutes les demandes du plan de contrôle de création ou de gestion de points d'accès multirégions doivent être routées vers la région `US West (Oregon)`. Pour les demandes de plan de données de points d'accès multirégions, il n'est pas nécessaire de spécifier les régions. Pour le plan de contrôle de basculement des points d'accès multirégions, la demande doit être routée vers l'une des cinq régions prises en charge. Pour plus d'informations sur les régions prises en charge par les points d'accès multirégionaux, consultezRestrictions et limitations des points d'accès multi-régions.

En outre, vous devez accorder l'`s3:ListAllMyBuckets`autorisation à l'utilisateur, au rôle ou à toute autre entité AWS Identity and Access Management (IAM) qui fait une demande de gestion d'un point d'accès multirégional.

Les exemples suivants montrent comment utiliser des points d'accès multi-régions avec des opérations compatibles dans Amazon S3.

Rubriques

- Compatibilité des points d'accès multirégionaux avec Services AWS les SDK AWS
- Compatibilité du point d'accès multi-régions avec les opérations S3
- Visualisez la configuration du routage de votre point d'accès multi-régions.
- Mettez à jour votre politique de compartiments Amazon S3 sous-jacente.
- Mise à jour de la configuration des itinéraires d'un point d'accès multi-régions
- Ajouter un objet à un compartiment dans votre point d'accès multirégion
- Récupération d'objets depuis votre point d'accès multirégion
- Liste des objets stockés dans un compartiment sous-jacent à votre point d'accès multirégion
- Utiliser une URL présignée avec des points d'accès multi-régions
- Utiliser un compartiment configuré avec Requester Pays avec des points d'accès multi-régions

Compatibilité des points d'accès multirégionaux avec Services AWS les SDK AWS

Pour utiliser un point d'accès multirégional avec des applications qui nécessitent un nom de compartiment Amazon S3, utilisez le nom de ressource Amazon (ARN) du point d'accès multirégional lorsque vous effectuez des demandes à l'aide d'un AWS SDK. Pour vérifier quels AWS SDK sont compatibles avec les points d'accès multirégionaux, consultez la section Compatibilité avec AWS les SDK.

Compatibilité du point d'accès multi-régions avec les opérations S3

Vous pouvez utiliser les opérations d'API du plan de données Amazon S3 suivantes pour effectuer des actions sur des objets dans des compartiments associés à votre point d'accès multirégion. Les opérations S3 suivantes peuvent accepter les ARN de points d'accès multi-régions :

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetObject
- GetObjectAcl
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging
- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- PutObjectAcl
- PutObjectLegalHold
- PutObjectRetention
- PutObjectTagging
- RestoreObject
- UploadPart

> **Note**
>
> Les points d'accès multirégionaux prennent en charge les opérations de copie utilisant des points d'accès multirégionaux uniquement comme destination lors de l'utilisation de l'ARN du point d'accès multirégional.

Vous pouvez utiliser les opérations de plan de contrôle Amazon S3 suivantes pour créer et gérer vos points d'accès multirégions :

- CreateMultiRegionAccessPoint
- DescribeMultiRegionAccessPointOperation
- GetMultiRegionAccessPoint
- GetMultiRegionAccessPointPolicy
- GetMultiRegionAccessPointPolicyStatus
- GetMultiRegionAccessPointRoutes
- ListMultiRegionAccessPoints
- PutMultiRegionAccessPointPolicy
- SubmitMultiRegionAccessPointRoutes

Visualisez la configuration du routage de votre point d'accès multi-régions.

AWS CLI

L'exemple de commande suivant récupère la configuration de l'itinéraire de votre point d'accès multi-régions afin que vous puissiez voir les statuts de routage actuels de vos compartiments. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
```

SDK for Java

Le kit SDK suivant pour le code Java récupère la configuration de l'itinéraire de votre point d'accès multi-régions afin que vous puissiez voir les statuts de routage actuels de vos compartiments. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider)
    .build();

GetMultiRegionAccessPointRoutesRequest request =
 GetMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .build();

GetMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.getMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Le SDK pour le JavaScript code suivant récupère la configuration de votre itinéraire de point d'accès multirégional afin que vous puissiez voir les statuts de routage actuels de vos buckets. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
const REGION = 'us-east-1'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new GetMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
      })
    )
```

```
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Le kit SDK suivant pour le code Python récupère la configuration des itinéraires de votre point d'accès multi-régions afin que vous puissiez voir les statuts de routage actuels de vos compartiments. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
s3.get_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Mettez à jour votre politique de compartiments Amazon S3 sous-jacente.

Pour accorder un accès approprié, vous devez également mettre à jour la politique sous-jacente du compartiment Amazon S3. Les exemples suivants délèguent le contrôle d'accès à la politique du point d'accès multi-régions. Une fois que vous avez délégué le contrôle d'accès à la politique de point d'accès multi-régions, la politique de compartiment n'est plus utilisée pour le contrôle d'accès lors de demandes effectuées via ce point d'accès.

Voici un exemple de politique de compartiment qui délègue le contrôle d'accès à la politique de point d'accès multi-régions. Pour utiliser cet exemple de politique de compartiment, remplacez *user input placeholders* par vos propres informations. Pour appliquer cette politique par le biais de la AWS CLI `put-bucket-policy` commande, comme indiqué dans l'exemple suivant, enregistrez la politique dans un fichier, par exemple`policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
```

```
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

L'exemple de commande `put-bucket-policy` suivant associe la politique de compartiment S3 mise à jour à votre compartiment S3 :

```
aws s3api put-bucket-policy
  --bucket DOC-EXAMPLE-BUCKET
  --policy file:///tmp/policy.json
```

Mise à jour de la configuration des itinéraires d'un point d'accès multi-régions

L'exemple de commande suivant met à jour la configuration des itinéraires du point d'accès multi-régions. Les commandes d'itinéraire des points d'accès multi-régions peuvent être exécutées dans les cinq régions suivantes :

- `ap-southeast-2`
- `ap-northeast-1`
- `us-east-1`
- `us-west-2`
- `eu-west-1`

Dans une configuration de routage de point d'accès multi-régions, vous pouvez définir des compartiments avec un statut de routage actif ou passif. Les compartiments actifs reçoivent du trafic, tandis que les compartiments passifs n'en reçoivent pas. Vous pouvez définir le statut de routage d'un compartiment en définissant la valeur `TrafficDialPercentage` de ce dernier sur `100` pour actif ou `0` pour passif.

AWS CLI

L'exemple de commande suivant met à jour la configuration du routage de votre point d'accès multi-régions. Dans cet exemple, la valeur *DOC-EXAMPLE-BUCKET1* est réglée sur le statut

actif et *DOC-EXAMPLE-BUCKET2* est réglée sur le statut passif. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=DOC-EXAMPLE-BUCKET1,TrafficDialPercentage=100
                Bucket=DOC-EXAMPLE-BUCKET2,TrafficDialPercentage=0
```

SDK for Java

Le kit SDK suivant pour le code Java met à jour la configuration de l'itinéraire de votre point d'accès multi-régions. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.ap-southeast-2)
    .credentialsProvider(credentialsProvider)
    .build();

SubmitMultiRegionAccessPointRoutesRequest request =
 SubmitMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .routeUpdates(
        MultiRegionAccessPointRoute.builder()
            .region("eu-west-1")
            .trafficDialPercentage(100)
            .build(),
        MultiRegionAccessPointRoute.builder()
            .region("ca-central-1")
            .bucket("111122223333")
            .trafficDialPercentage(0)
            .build()
    )
    .build();

SubmitMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.submitMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Le SDK de JavaScript code suivant met à jour la configuration de votre itinéraire de point d'accès multirégional. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'DOC-EXAMPLE-BUCKET1',
            TrafficDialPercentage: 0,
          },
        ],
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Le kit SDK suivant pour le code Python met à jour la configuration de l'itinéraire de votre point d'accès multi-régions. Pour utiliser cet exemple de syntaxe, remplacez *user input placeholders* par vos propres informations.

```
s3.submit_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap,
        RouteUpdates= [{
            'Bucket': DOC-EXAMPLE-BUCKET,
            'Region': ap-southeast-2,
            'TrafficDialPercentage': 10
        }])
```

Ajouter un objet à un compartiment dans votre point d'accès multirégion

Pour ajouter un objet au compartiment associé au point d'accès multirégion, vous pouvez utiliser l'opération PutObject. Pour maintenir synchronisés tous les compartiments dans le point d'accès multirégion, activez la réplication entre régions.

> **Note**
>
> Pour utiliser cette opération, vous devez disposer de l'autorisation `s3:PutObject` pour le point d'accès multirégion. Pour plus d'informations sur les exigences relatives aux autorisations des points d'accès multirégions, consultez Autorisations.

AWS CLI

L'exemple de demande de plan de données suivant charge *example.txt* dans le point d'accès multirégion spécifié. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
aws s3api put-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --key example.txt --
body example.txt
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();

PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
        .key("example.txt")
        .build();

s3Client.putObject(objectRequest, RequestBody.fromString("Hello S3!"));
```

SDK for JavaScript

```
const client = new S3Client({});

async function putObjectExample() {
    const command = new PutObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt",
        Body: "Hello S3!",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.put_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt',
    Body='Hello S3!'
)
```

Récupération d'objets depuis votre point d'accès multirégion

Pour récupérer des objets depuis le point d'accès multirégion, vous pouvez utiliser l'opération GetObject.

> **Note**
>
> Pour utiliser cette opération d'API, vous devez disposer de l'autorisation `s3:GetObject` pour le point d'accès multirégion. Pour plus d'informations sur les exigences relatives aux autorisations des points d'accès multirégions, consultez Autorisations.

AWS CLI

L'exemple de demande de plan de données suivant récupère *`example.txt`* depuis le point d'accès multirégion spécifié et le télécharge en tant que *`downloaded_example.txt`*. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3api get-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --
key example.txt downloaded_example.txt
```

SDK for Java

```
S3Client s3 = S3Client
   .builder()
   .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.getObject(getObjectRequest);
```

SDK for JavaScript

```
const client = new S3Client({})

async function getObjectExample() {
```

```
    const command = new GetObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt"
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.get_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt'
)
```

Liste des objets stockés dans un compartiment sous-jacent à votre point d'accès multirégion

Pour renvoyer la liste des objets stockés dans un compartiment sous-jacent à votre point d'accès multirégion, utilisez l'opération ListObjectsV2. Dans l'exemple de commande suivant, tous les objets du point d'accès multirégion spécifié sont répertoriés à l'aide de l'ARN du point d'accès multirégion. Dans ce cas, l'ARN du point d'accès multirégion est :

`arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap`

> **Note**
>
> Pour utiliser cette opération d'API, vous devez disposer de l'autorisation `s3:ListBucket` pour le point d'accès multirégion et le compartiment sous-jacent. Pour plus d'informations sur les exigences relatives aux autorisations des points d'accès multirégions, consultez Autorisations.

AWS CLI

L'exemple de demande de plan de données suivant répertorie les objets présents dans le compartiment sous-jacent au point d'accès multirégion spécifié par l'ARN. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
aws s3api list-objects-v2 --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();

String bucketName = "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap";

ListObjectsV2Request listObjectsRequest = ListObjectsV2Request
    .builder()
    .bucket(bucketName)
    .build();

 s3Client.listObjectsV2(listObjectsRequest);
```

SDK for JavaScript

```
const client = new S3Client({});

async function listObjectsExample() {
    const command = new ListObjectsV2Command({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.list_objects_v2(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap'
)
```

Utiliser une URL présignée avec des points d'accès multi-régions

Vous pouvez utiliser une URL présignée pour générer une URL permettant aux autres d'accéder à vos compartiments Amazon S3 via un point d'accès multirégion Amazon S3. Lorsque vous créez une URL présignée, vous l'associez à une action d'objet spécifique, telle qu'un chargement S3 (`PutObject`) ou un téléchargement S3 (`GetObject`). Vous pouvez partager l'URL présignée, et toute personne y ayant accès peut effectuer l'action intégrée dans l'URL comme s'il s'agissait de l'utilisateur connecté d'origine.

Les URL présignées ont une date d'expiration. Lorsque le délai d'expiration est atteint, l'URL ne fonctionne plus.

Avant d'utiliser des points d'accès multi-régions S3 avec des URL présignées, vérifiez la compatibilité du kit SDK AWS avec l'algorithme SigV4A. Vérifiez que la version de votre kit SDK prend en charge SigV4A comme mise en œuvre de la signature qui est utilisée pour signer les requêtes globales Région AWS . Pour plus d'informations sur l'utilisation d'URL présignées avec Amazon S3, consultez Partage d'objets à l'aide d'URL présignées.

Les exemples suivants montrent comment vous pouvez utiliser les points d'accès multirégions avec des URL présignées. Pour utiliser ces exemples, remplacez *`user input placeholders`* par vos propres informations.

AWS CLI

```
aws s3 presign
 arn:aws:s3::123456789012:accesspoint/MultiRegionAccessPoint_alias/example-file.txt
```

SDK for Python

```
import logging
import boto3
```

```
from botocore.exceptions import ClientError

s3_client = boto3.client('s3',aws_access_key_id='xxx',aws_secret_access_key='xxx')
s3_client.generate_presigned_url(HttpMethod='PUT',ClientMethod="put_object",
 Params={'Bucket':'arn:aws:s3::123456789012:accesspoint/
abcdef0123456.mrap','Key':'example-file'})
```

SDK for Java

```
S3Presigner s3Presigner = S3Presigner.builder()
    .credentialsProvider(StsAssumeRoleCredentialsProvider.builder()
        .refreshRequest(assumeRole)
        .stsClient(stsClient)
        .build())
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example-file")
    .build();

GetObjectPresignRequest preSignedReq = GetObjectPresignRequest.builder()
    .getObjectRequest(getObjectRequest)
    .signatureDuration(Duration.ofMinutes(10))
    .build();

PresignedGetObjectRequest presignedGetObjectRequest =
 s3Presigner.presignGetObject(preSignedReq);
```

Note

Pour utiliser SigV4A avec des informations d'identification de sécurité temporaires, par exemple, lorsque vous utilisez des rôles IAM, assurez-vous de demander les informations d'identification temporaires à un point de terminaison régional situé dans AWS Security Token Service (), plutôt qu'à un point de terminaison global.AWS STS Si vous utilisez le point de terminaison global pour AWS STS (`sts.amazonaws.com`), AWS STS cela générera des informations d'identification temporaires à partir d'un point de terminaison global, ce qui n'est pas pris en charge par Sig4A. Par conséquent, vous obtenez une erreur. Pour résoudre ce problème, utilisez l'un des points de terminaison régionaux répertoriés pour AWS STS.

Utiliser un compartiment configuré avec Requester Pays avec des points d'accès multi-régions

Si un compartiment S3 associé à vos points d'accès multirégions est configuré pour utiliser le paiement par le demandeur, le demandeur paiera à la fois la demande de compartiment, le téléchargement et tous les coûts liés aux points d'accès multirégions. Pour plus d'informations, consultez Tarification Amazon S3.

Voici un exemple de demande de plan de données adressée à un point d'accès multirégion connecté à un compartiment de type Paiement par le demandeur.

AWS CLI

Pour télécharger des objets à partir d'un point d'accès multirégion connecté à un compartiment de type Paiement par le demandeur, vous devez spécifier `--request-payer requester` dans le cadre de votre demande get-object. Vous devez également spécifier le nom du fichier dans le compartiment, ainsi que l'emplacement où le fichier téléchargé doit être stocké.

```
aws s3api get-object --bucket MultiRegionAccessPoint_ARN --request-payer requester
 --key example-file-in-bucket.txt example-location-of-downloaded-file.txt
```

SDK for Java

Pour télécharger des objets à partir d'un point d'accès multirégion connecté à un compartiment de type Paiement par le demandeur, vous devez spécifier `RequestPayer.REQUESTER` dans le cadre de votre demande `GetObject`. Vous devez également spécifier le nom du fichier dans le compartiment, ainsi que l'emplacement où il doit être stocké.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Surveillance et journalisation des demandes effectuées via un point d'accès multi-Régions vers les ressources sous-jacentes

Amazon S3 journalise les demandes effectuées via les points d'accès multi-régions et les demandes adressées aux opérations d'API qui les gèrent, telles que `CreateMultiRegionAccessPoint` et `GetMultiRegionAccessPointPolicy`. Les requêtes adressées à Amazon S3 via un point d'accès apparaissent dans vos journaux d'accès au serveur S3 et les journaux AWS CloudTrail portant le nom d'hôte du point d'accès multi-Régions. Le nom d'hôte d'un point d'accès prend la forme *`MRAP_alias`*`.accesspoint.s3-global.amazonaws.com`. Par exemple, supposons que vous disposez de la configuration de compartiment et de point d'accès multi-Régions suivante :

- Un compartiment nommé `my-bucket-usw2` dans la région `us-west-2` qui contient un objet `my-image.jpg`.
- Un compartiment nommé `my-bucket-aps1` dans la région `ap-south-1` qui contient un objet `my-image.jpg`.
- Un compartiment nommé `my-bucket-euc1` dans la région `eu-central-1` qui ne contient pas d'objet nommé `my-image.jpg`.
- Un point d'accès multi-Régions nommé `my-mrap`avec l'alias `mfzwi23gnjvgw.mrap` qui est configuré pour traiter les demandes des trois compartiments.
- L'ID de compte AWS est `123456789012`.

Une demande effectuée pour récupérer `my-image.jpg` directement via le compartiment apparaît dans vos journaux avec un nom d'hôte *`bucket_name`*`.s3.`*`Region`*`.amazonaws.com`.

Si vous faites la demande par le biais du point d'accès multi-régions, Amazon S3 détermine d'abord lequel des compartiments des différentes régions est le plus proche. Une fois qu'Amazon S3 aura déterminé le compartiment à utiliser pour traiter la demande, il enverra la demande à ce compartiment et journalisera l'opération à l'aide du nom d'hôte du point d'accès multi-régions. Dans cet exemple, si Amazon S3 relaye la demande à `my-bucket-aps1`, vos journaux refléteront une demande `GET` réussie pour `my-image.jpg` depuis `my-bucket-aps1`, en utilisant un nom d'hôte de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

> **Important**
>
> Les points d'accès multi-régions ne connaissent pas le contenu des données des compartiment sous-jacents. Par conséquent, le compartiment qui reçoit la demande peut

ne pas contenir les données demandées. Par exemple, si Amazon S3 détermine que le compartiment `my-bucket-euc1` est le plus proche, vos journaux indiqueront l'échec d'une demande `GET` pour `my-image.jpg`, depuis `my-bucket-euc1`, avec un nom d'hôte de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Si la demande a été acheminée vers `my-bucket-usw2` au lieu de cela, vos journaux indiqueraient une demande `GET` réussie.

Pour en savoir plus sur les journaux d'accès au serveur Amazon S3, consultez Enregistrement de demandes avec journalisation des accès au serveur. Pour plus d'informations sur AWS CloudTrail, consultez Qu'est-ce que AWS CloudTrail ? dans le guide de l'utilisateur AWS CloudTrail.

Surveillance et journalisation des demandes faites aux opérations d'API de gestion de points d'accès multi-régions

Amazon S3 fournit plusieurs opérations d'API de gestion des points d'accès multi-régions, telles que `CreateMultiRegionAccessPoint` et `GetMultiRegionAccessPointPolicy`. Lorsque vous effectuez des demandes à ces opérations d'API à l'aide de l'AWS Command Line Interface (AWS CLI), des kits SDK AWS ou de l'API REST d'Amazon S3, Amazon S3 traite ces demandes de manière asynchrone. Si vous disposez des autorisations appropriées pour la demande, Amazon S3 renverra un jeton pour ces demandes. Vous pouvez utiliser ce jeton avec `DescribeAsyncOperation` pour vous aider à afficher le statut des opérations asynchrones en cours. Amazon S3 traite `DescribeAsyncOperation` de manière synchrone. Pour afficher le statut des demandes asynchrones, vous pouvez utiliser la console Amazon S3, l'AWS CLI, les kits SDK ou l'API REST.

> **Note**
>
> La console affiche uniquement l'état des demandes asynchrones effectuées au cours des 14 jours précédents. Pour afficher l'état des demandes plus anciennes, utilisez la AWS CLI, les kits SDK ou l'API REST.

Les opérations de gestion asynchrone peuvent avoir l'un des états suivants :

NEW

Amazon S3 a reçu la demande et se prépare à effectuer l'opération.

IN_PROGRESS

Amazon S3 effectue actuellement l'opération.

SUCCESS

L'opération a réussi. La réponse inclut des informations pertinentes, telles que l'alias de point d'accès multi-Régions pour une demande `CreateMultiRegionAccessPoint`.

FAILED

L'opération a échoué. La réponse inclut un message d'erreur indiquant la raison de l'échec de la demande.

Utilisation d'AWS CloudTrail avec des points d'accès multi-régions

Vous pouvez utiliser AWS CloudTrail pour afficher, rechercher, télécharger, archiver, analyser et traiter l'activité de compte sur votre infrastructure AWS. Grâce aux points d'accès multi-régions et à la journalisation CloudTrail, vous pouvez identifier les éléments suivants :

- Qui ou quoi a pris quelle mesure
- Quelles ressources ont été utilisées
- Moment où l'événement est survenu
- Autres détails concernant l'événement

Vous pouvez utiliser ces informations de journalisation pour vous aider à analyser les activités qui se sont produites via vos points d'accès multi-régions et à y répondre.

Comment configurer AWS CloudTrail pour les points d'accès multi-Régions

Pour activer la journalisation CloudTrail de toutes les opérations liées à la création ou à la maintenance de points d'accès multi-régions, vous devez configurer la journalisation CloudTrail pour enregistrer les événements dans la région USA Ouest (Oregon). Vous devez configurer votre journalisation de cette façon, quelle que soit la région dans laquelle vous vous trouvez lorsque vous effectuez la demande, ou quelles que soient les régions prises en charge par le point d'accès multi-régions. Toutes les demandes de création ou de gestion d'un point d'accès multi-régions sont acheminées via la région USA Ouest (Oregon). Nous vous recommandons d'ajouter cette région à un journal d'activité existant ou de créer un journal d'activité contenant cette région et toutes les régions associées au point d'accès multi-régions.

Amazon S3 enregistre les demandes de journalisation effectuées via un point d'accès multi-régions et les demandes adressées aux opérations d'API qui gèrent les points d'accès, telles que `CreateMultiRegionAccessPoint` et `GetMultiRegionAccessPointPolicy`. Lorsque vous enregistrez ces demandes via un point d'accès multi-Régions, elles apparaissent dans votre AWS CloudTrail avec le nom d'hôte du point d'accès multi-Régions. Par exemple, si vous effectuez des demandes à un compartiment via un point d'accès multi-régions avec l'alias `mfzwi23gnjvgw.mrap`, les entrées du journal CloudTrail auront un nom d'hôte de `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Pour cette raison, lorsque vous consulterez les journaux CloudTrail pour un point d'accès multi-régions, vous verrez des demandes effectuées aux compartiments sous-jacents. Certaines de ces demandes peuvent être des demandes directes vers le compartiment et non acheminées via le point d'accès multi-régions. Gardez ce fait en tête lorsque vous examinez le trafic. Si un compartiment est dans un point d'accès multi-Régions, les demandes pourront toujours être adressées directement à ce compartiment sans passer par le point d'accès multi-Régions.

Des événements asynchrones sont impliqués dans la création et la gestion des points d'accès multi-Régions. Les demandes asynchrones ne sont pas accompagnées d'événements d'achèvement dans le journal CloudTrail. Pour en savoir plus amples sur les demandes asynchrones, consultez Surveillance et journalisation des demandes faites aux opérations d'API de gestion de points d'accès multi-régions.

Pour plus d'informations sur AWS CloudTrail, consultez Qu'est-ce que AWS CloudTrail ? dans le guide de l'utilisateur AWS CloudTrail.

Sécurité Amazon S3

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

Sécurité du cloud

AWS est chargé de protéger l'infrastructure qui gère AWS les services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. L'efficacité de notre sécurité est régulièrement testée et vérifiée par des auditeurs tiers dans le cadre des programmes de conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Simple Storage Service (Amazon S3), veuillez consulter Services AWS concernés par le programme de conformité.

Sécurité dans le cloud

Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation ainsi que les lois et réglementations applicables. Pour Amazon S3, votre responsabilité inclut les domaines suivants :

- Gestion de vos données, y compris propriété de l'objet et chiffrement.
- Classification de vos ressources.
- Gestion des accès à vos données à l'aide de rôles IAM et d'autres configurations de service pour appliquer les autorisations appropriées.
- Activation de contrôles de détection tels que AWS CloudTrailAmazon GuardDuty pour Amazon S3.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lorsque vous utilisez Simple Storage Service (Amazon S3). Les rubriques suivantes vous montrent comment configurer Amazon S3 pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui peuvent vous aider à surveiller et à sécuriser vos ressources Amazon S3.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Protection des données dans Amazon S3

Simple Storage Service (Amazon S3) offre une infrastructure de stockage hautement durable, pensée pour le stockage de données primaires et stratégiques. S3 standard, S3 Intelligent-Tiering, S3 standard – Accès peu fréquent, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive stockent des objets de façon redondante sur plusieurs appareils sur un minimum de trois zones de disponibilité dans une Région AWS. Une zone de disponibilité est un ou plusieurs centres de données discrets dotés d'une alimentation, d'un réseau et d'une connectivité redondants dans une Région AWS. Les zones de disponibilité sont physiquement séparées par une distance significative, de plusieurs kilomètres, de toute autre zone de disponibilité, bien qu'elles se trouvent toutes à moins de 100 km (60 miles) les unes des autres. La classe de stockage S3 unizone – Accès peu fréquent stocke les données de façon redondante sur plusieurs appareils au

sein d'une seule zone de disponibilité. Ces services sont conçus pour gérer les défaillances de périphériques concurrents en détectant et en réparant rapidement toute redondance perdue, et ils vérifient également régulièrement l'intégrité de vos données à l'aide de totaux de contrôle.

Le stockage standard Simple Storage Service (Amazon S3) présente les caractéristiques suivantes :

- Soutenu par l'accord de niveau de service contrat de niveau de service (SLA) Amazon S3.
- Conçu pour fournir une 99,999999999 % de durabilité et 99,99 % de disponibilité des objets sur une année donnée.
- S3 standard, S3 Intelligent-Tiering, S3 standard – Accès peu fréquent, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont tous conçus pour garder les données en cas de perte d'une zone de disponibilité Amazon S3 complète.

Simple Storage Service (Amazon S3) protège également vos données à l'aide de la gestion des versions. Vous pouvez utiliser la gestion des versions pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans votre compartiment Amazon S3. Le contrôle de version permet de récupérer facilement les données en cas d'actions involontaires des utilisateurs ou de défaillances des applications. Par défaut, les demandes récupèrent la version écrite la plus récente. Vous pouvez récupérer les versions antérieures d'un objet en spécifiant une version de l'objet dans une demande.

Outre la gestion des versions S3, vous pouvez également utiliser le verrouillage d'objets Amazon S3 et la réplication S3 pour protéger vos données. Pour plus d'informations, consultez le Tutoriel : protection des données sur Amazon S3 contre les suppressions accidentelles ou les bogues d'application à l'aide de la gestion des versions S3, du verrouillage d'objets S3 et de la réplication S3.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer des comptes utilisateur individuels AWS Identity and Access Management, afin que chaque utilisateur ne dispose que des autorisations nécessaires pour accomplir ses tâches.

Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d’informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l’information fédérale).

Les bonnes pratiques de sécurité suivantes s'appliquent également à la protection des données dans Amazon S3 :

- Implement server-side encryption
- Enforce encryption of data in transit
- Consider using Macie with Amazon S3
- Identify and audit all your Amazon S3 buckets
- Monitor Amazon Web Services security advisories

Protection des données à l'aide du chiffrement

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

La protection des données fait référence à la protection des données pendant que celles-ci sont en transit (à destination ou en provenance d'Amazon S3) et au repos (durant leur stockage sur les disques de centres de données Amazon S3). Vous pouvez protéger les données en transit à l'aide du protocole Secure Socket Layer/Transport Layer Security (SSL/TLS) ou du chiffrement côté client. Pour protéger des données au repos dans Amazon S3, vous disposer des options suivantes :

- Chiffrement côté serveur : Amazon S3 chiffre vos objets avant de les enregistrer sur les disques des centres de AWS données, puis les déchiffre lorsque vous les téléchargez.

  Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à

utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

Pour plus d'informations sur chaque option pour le chiffrement côté serveur, consultez Protection des données avec le chiffrement côté serveur.

Pour configurer le chiffrement côté serveur, consultez :

- Spécification du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
- Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS)
- the section called "Spécification de DSSE-KMS"
- Spécification du chiffrement côté serveur avec des clés fournies par le client (SSE-C)

- Chiffrement côté client : vous chiffrez vos données côté client et chargez les données chiffrées dans Amazon S3. Dans ce cas, vous gérez le processus de chiffrement, les clés de chiffrement et les outils associés.

  Pour configurer le chiffrement côté client, veuillez consulter Protection des données avec le chiffrement côté client.

Pour savoir quel pourcentage de vos octets de stockage sont chiffrés, vous pouvez utiliser les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec S3 Storage Lens. Pour obtenir la liste complète des métriques, consultez le Glossaire des métriques S3 Storage Lens.

Pour plus d'informations sur le chiffrement côté serveur et le chiffrement côté client, consultez les rubriques suivantes.

Rubriques

Protection des données avec le chiffrement côté serveur

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Le chiffrement côté serveur est le chiffrement des données à leur destination par l'application ou le service qui les reçoit. Amazon S3 chiffre vos données au niveau de l'objet lorsqu'il les écrit sur les disques des centres de AWS données et les déchiffre pour vous lorsque vous y accédez. Tant que vous authentifiez votre demande et que vous avez des autorisations d'accès, il n'y a aucune différence dans la manière dont vous accédez aux objets chiffrés ou déchiffrés. Par exemple, si vous partagez vos objets en utilisant une URL pré-signée, cette URL fonctionne de la même manière pour les objets chiffrés et déchiffrés. En outre, quand vous répertoriez les objets de votre compartiment, les opérations d'API de liste renvoient la liste de tous les objets, qu'ils soient chiffrés ou non.

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-

KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

> **Note**
>
> Vous ne pouvez pas appliquer simultanément différents types de chiffrement côté serveur au même objet.

Si vous devez chiffrer vos objets existants, utilisez S3 Batch Operations et S3 Inventory. Pour plus d'informations, consultez Encrypting objects with Amazon S3 Batch Operations (Chiffrement d'objets avec des opérations par lot Amazon S3) et Exécution des opérations par lot à grande échelle sur des objets Amazon S3.

Vous avez le choix entre quatre options de chiffrement côté serveur, qui s'excluent mutuellement. Votre choix dépendra de la façon dont vous décidez de gérer les clés de chiffrement et du nombre de couches de chiffrement que vous souhaitez appliquer.

Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3. L'option par défaut pour le chiffrement côté serveur est avec des clés gérées Amazon S3 (SSE-S3). Chaque objet est chiffré à l'aide d'une clé unique. Comme protection supplémentaire, SSE-S3 chiffre la clé elle-même à l'aide d'une clé racine dont il effectue une rotation régulière. SSE-S3 utilise l'un des chiffrements par bloc les plus puissants qui existent, Advanced Encryption Standard 256 bits (AES-256), pour chiffrer vos données. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS)

Le chiffrement côté serveur avec AWS KMS keys (SSE-KMS) est fourni par le biais d'une intégration du service à AWS KMS Amazon S3. Avec AWS KMS, vous avez plus de contrôle sur vos clés. Par exemple, vous pouvez afficher des clés distinctes, modifier des politiques de contrôle et suivre les clés dans AWS CloudTrail. En outre, vous pouvez créer et gérer les clés gérées par le client ou utiliser les Clés gérées par AWS qui sont propres à vous-même, à votre service et à votre région. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

Chiffrement double couche côté serveur avec clés AWS Key Management Service (AWS KMS) (DSSE-KMS)

Le chiffrement double couche côté serveur avec AWS KMS keys (DSSE-KMS) est similaire au SSE-KMS, mais le DSSE-KMS applique deux couches individuelles de chiffrement au niveau des objets au lieu d'une seule couche. Les deux couches de chiffrement étant appliquées à un objet côté serveur, vous pouvez utiliser un large éventail d' Services AWS outils pour analyser les données dans S3 tout en utilisant une méthode de chiffrement répondant à vos exigences de conformité. Pour plus d'informations, consultez Utilisation du chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS).

Chiffrement côté serveur avec clés fournies par le client (SSE-C)

Dans le cadre du chiffrement côté serveur avec des clés fournies par le client (SSE-C), vous gérez les clés de chiffrement, et Amazon S3 gère le chiffrement à mesure qu'il écrit les données sur les disques et le déchiffrement au moment où vous accédez à vos objets. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C).

Amazon S3 chiffre désormais automatiquement tous les nouveaux objets.

Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. SSE-S3, qui utilise la norme de chiffrement avancé 256 bits (AES-256), est automatiquement appliqué à tous les nouveaux compartiments et à tout compartiment S3 existant qui n'a pas déjà un chiffrement par défaut configuré. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans les AWS CloudTrail journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans le AWS Command Line Interface (AWS CLI) et les AWS SDK.

Les sections suivantes répondent aux questions concernant cette mise à jour.

Amazon S3 modifie-t-il les paramètres de chiffrement par défaut de mes compartiments existants pour lesquels le chiffrement par défaut est déjà configuré ?

Non Aucune modification n'est apportée à la configuration de chiffrement par défaut pour un compartiment existant sur lequel le chiffrement SSE-S3 ou le chiffrement côté serveur avec AWS Key

Management Service (AWS KMS) clés (SSE-KMS) sont déjà configurés. Pour plus d'informations sur la manière de définir le comportement de chiffrement par défaut pour les compartiments, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3. Pour plus d'informations sur les paramètres de chiffrement de SSE-S3 et SSE-KMS, consultez Protection des données avec le chiffrement côté serveur.

Le chiffrement par défaut est-il activé sur mes compartiments existants qui n'ont pas de chiffrement par défaut configuré ?

Oui. Amazon S3 configure désormais le chiffrement par défaut de tous les compartiments non chiffrés existants pour appliquer le chiffrement côté serveur avec les clés gérées S3 (SSE-S3) comme niveau de base du chiffrement pour les nouveaux objets chargés dans ces compartiments. Les objets qui se trouvent déjà dans un compartiment non chiffré existant ne seront pas automatiquement chiffrés.

Comment puis-je afficher l'état de chiffrement par défaut des nouveaux chargements d'objets ?

À l'heure actuelle, vous pouvez consulter l'état de chiffrement par défaut des nouveaux objets chargés dans AWS CloudTrail les journaux, S3 Inventory et S3 Storage Lens, sur la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans le AWS Command Line Interface (AWS CLI) et les AWS SDK.

- Pour consulter vos CloudTrail événements, reportez-vous à la section Affichage CloudTrail des événements dans la CloudTrail console du Guide de AWS CloudTrail l'utilisateur. CloudTrail les journaux fournissent le suivi des API `PUT` et des `POST` requêtes adressées à Amazon S3. Lorsque le chiffrement par défaut est utilisé pour chiffrer des objets dans vos compartiments, les CloudTrail journaux `PUT` et les demandes d'`POST`API incluront le champ suivant comme paire nom-valeur :. `"SSEApplied":"Default_SSE_S3"`
- Pour afficher l'état de chiffrement automatique des nouveaux objets chargés dans S3 Inventory, configurez un rapport S3 Inventory pour inclure le champ de métadonnées Encryption (Chiffrement), puis consultez l'état de chiffrement de chaque nouvel objet dans le rapport. Pour plus d’informations, consultez Configuration d’Amazon S3 Inventory.
- Pour consulter l'état de chiffrement automatique des nouveaux chargements d'objets dans S3 Storage Lens, configurez un tableau de bord S3 Storage Lens et consultez les mesures relatives à Encrypted bytes (Octets chiffrés) et Encrypted object count (Nombre d'objets chiffrés) dans la catégorie Data protection (Protection des données) du tableau de bord. Pour plus d’informations, consultez Créer un tableau de bord Amazon S3 Storage Lens et Afficher les métriques S3 Storage Lens sur les tableaux de bord.

- Pour afficher le statut du chiffrement automatique au niveau du compartiment dans la console Amazon S3, vérifiez le chiffrement par défaut de vos compartiments Amazon S3 dans la console Amazon S3. Pour plus d'informations, consultez Configuration du chiffrement par défaut.
- Pour afficher l'état du chiffrement automatique sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans le AWS Command Line Interface (AWS CLI) et les AWS SDK, vérifiez l'en-tête de réponse `x-amz-server-side-encryption` lorsque vous utilisez des API d'action sur des objets, telles que PutObjectet GetObject.

Que dois-je faire pour profiter de ce changement ?

Vous n'êtes pas tenu d'apporter des modifications à vos applications existantes. Le chiffrement par défaut étant activé pour tous vos compartiments, tous les nouveaux objets chargés sur Amazon S3 sont automatiquement chiffrés.

Puis-je désactiver le chiffrement des nouveaux objets qui sont enregistrés dans mon compartiment ?

Non. SSE-S3 est le nouveau niveau de base du chiffrement qui est appliqué à tous les nouveaux objets chargés dans votre compartiment. Vous ne pouvez plus désactiver le chiffrement pour les nouveaux chargements d'objets.

Y aura-t-il une incidence sur mes frais ?

Non. Le chiffrement par défaut avec SSE-S3 est disponible sans coût supplémentaire. Vous serez facturé pour le stockage, les requêtes et les autres fonctionnalités de S3, comme d'habitude. Pour en savoir plus sur la tarification, veuillez consulter Tarification Amazon S3.

Amazon S3 va-t-il chiffrer mes objets existants qui ne sont pas chiffrés ?

Non. À partir du 5 janvier 2023, Amazon S3 ne chiffrera automatiquement que les nouveaux chargements d'objets. Pour chiffrer des objets existants, vous pouvez utiliser les opérations par lots S3 pour créer des copies chiffrées de vos objets. Ces copies chiffrées conserveront les données et le nom de l'objet existant et seront chiffrées à l'aide des clés de chiffrement que vous spécifiez. Pour plus de détails, consultez Encrypting objects with Amazon S3 Batch Operations (Chiffrement d'objets avec les opérations par lot Amazon S3) dans le blog sur le stockage AWS .

Je n'ai pas activé le chiffrement pour mes compartiments avant cette version. Dois-je modifier la façon dont j'accède aux objets ?

Non. Le chiffrement par défaut avec SSE-S3 chiffre automatiquement vos données lorsqu'elles sont enregistrées sur Amazon S3 et les déchiffre pour vous lorsque vous y accédez. Il n'y a aucun changement dans la façon dont vous accédez aux objets qui sont automatiquement chiffrés.

Dois-je modifier la façon dont j'accède à mes objets chiffrés côté client ?

Non. Tous les objets chiffrés côté client qui sont chiffrés avant d'être chargés dans Amazon S3 arrivent sous forme d'objets texte chiffré dans Amazon S3. Ces objets seront désormais dotés d'une couche supplémentaire de chiffrement SSE-S3. Vos charges de travail qui utilisent des objets chiffrés côté client ne nécessiteront aucune modification de vos services clients ou de vos paramètres d'autorisation.

> **Note**
>
> HashiCorp Les utilisateurs de Terraform qui n'utilisent pas de version mise à jour du AWS fournisseur peuvent constater une dérive inattendue après avoir créé de nouveaux compartiments S3 sans configuration de chiffrement définie par le client. Pour éviter cette dérive, mettez à jour la version de votre AWS fournisseur Terraform vers l'une des versions suivantes : n'importe quelle 4.x version3.76.1, ou. 2.70.4

Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Tous les nouveaux chargements d'objets dans les compartiments Amazon S3 sont chiffrés par défaut à l'aide du chiffrement côté serveur avec les clés gérées Amazon S3 (SSE-S3).

Le chiffrement côté serveur protège les données au repos. Amazon S3 chiffre chaque objet à l'aide d'une clé unique. Comme protection supplémentaire, il chiffre la clé elle-même à l'aide d'une clé dont il effectue une rotation régulière. Le chiffrement côté serveur Amazon S3 utilise AES-GCM (Advanced Encryption Standard Galois/Counter Mode) 256 bits pour chiffrer tous les objets chargés.

L'utilisation du chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) n'entraîne pas de frais supplémentaires. Toutefois, les demandes de configuration de la fonction de chiffrement par défaut seront facturées comme des demandes Amazon S3 standard. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3.

Si vous souhaitez que vos chargements de données soient chiffrés à l'aide de clés gérées uniquement par Amazon S3, vous pouvez utiliser la politique de compartiment suivante. Par exemple, la stratégie de compartiment suivante refuse les autorisations de charger un objet si la demande n'inclut pas l'en-tête `x-amz-server-side-encryption` demandant le chiffrement côté serveur :

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
      }
    }
   ]
}
```

> **Note**
>
> Un chiffrement côté serveur chiffre uniquement les données d'objet, pas les métadonnées d'objet.

Prise en charge de l'API pour le chiffrement côté serveur

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

Pour configurer un chiffrement côté serveur à l'aide des API REST de création d'objet, vous devez fournir l'en-tête de demande `x-amz-server-side-encryption`. Pour en savoir plus sur les API REST, consultez Utilisation de l'API REST.

Les API Amazon S3 suivantes prennent en charge cet en-tête :

- Opérations PUT : spécifiez l'en-tête de demande lors du chargement des données grâce à l'API PUT. Pour plus d’informations, consultez Objet PUT.
- Lancement du chargement partitionné : spécifiez l'en-tête dans la demande initiale lors du chargement d'objets volumineux grâce à l'API de chargement partitionné. Pour plus d’informations, consultez Lancement du chargement partitionné.
- Opérations COPY : lorsque vous copiez un objet, vous disposez à la fois d'un objet source et d'un objet cible. Pour plus d’informations, consultez Objet PUT - Copy.

> **Note**
>
> Lors de l'utilisation d'une opération POST pour charger un objet, à la place de l'en-tête de demande, vous fournissez les mêmes informations dans les champs du formulaire. Pour plus d’informations, consultez Objet POST.

Les AWS SDK fournissent également des API wrapper que vous pouvez utiliser pour demander un chiffrement côté serveur. Vous pouvez également utiliser le AWS Management Console pour télécharger des objets et demander un chiffrement côté serveur.

Pour obtenir des informations plus générales, consultez Concepts AWS KMS dans le Guide du développeur AWS Key Management Service .

Rubriques

- Spécification du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)

Spécification du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre

configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

Vous pouvez spécifier SSE-S3 à l'aide de la console S3, des API REST, AWS des SDK et AWS Command Line Interface ().AWS CLI Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Utilisation de la console S3

Cette rubrique décrit comment définir ou modifier le type de chiffrement qu'un objet utilise à l'aide de la AWS Management Console. Lorsque vous copiez un objet en utilisant la console, Amazon S3 copie l'objet en l'état. Cela signifie que si l'objet source est chiffré, l'objet cible est également chiffré. La console vous permet d'ajouter ou de modifier le chiffrement d'un objet.

> **Note**
>
> - Si vous modifiez le chiffrement d'un objet, un nouvel objet est créé pour remplacer l'ancien. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Le rôle qui modifie la propriété devient également le propriétaire du nouvel objet ou (version de l'objet).
> - Si vous modifiez le type de chiffrement d'un objet doté de balises définies par l'utilisateur, vous devez disposer de cette `s3:GetObjectTagging` autorisation. Si vous modifiez le type de chiffrement d'un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`s3:GetObjectTagging`autorisation.
>
>   Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, le type de chiffrement de l'objet sera mis à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

Pour modifier le chiffrement d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.

4. Dans la liste Objets, choisissez le nom de l'objet pour lequel vous souhaitez ajouter ou modifier le chiffrement.

   La page de détails de l'objet apparaît, avec plusieurs sections qui affichent les propriétés de votre objet.
5. Choisissez l'onglet Propriétés.
6. Faites défiler la page jusqu'à la section Paramètres de chiffrement côté serveur, puis choisissez Modifier.
7. Sous Paramètres de chiffrement, choisissez Utiliser les paramètres de chiffrement par défaut du compartiment ou Ignorer les paramètres de chiffrement par défaut du compartiment.
8. Si vous avez choisi Ignorer les paramètres de chiffrement par défaut du compartiment, configurez les paramètres de chiffrement suivants.

   - Sous Type de chiffrement, choisissez Clés gérées par Amazon S3 (SSE-S3). SSE-S3 utilise l'un des chiffrements par bloc les plus puissants qui existent, Advanced Encryption Standard à 256 bits (AES-256) pour chiffrer chaque objet. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).
9. Choisissez Enregistrer les modifications.

> **Note**
>
> Cette action applique le chiffrement à tous les objets spécifiés. Lorsque vous chiffrez des dossiers, attendez la fin de l'opération d'enregistrement pour ajouter de nouveaux objets au dossier.

Utilisation de l'API REST

Lors de la création d'un objet (c'est-à-dire lorsque vous chargez un nouvel objet ou effectuez une copie d'un objet existant), vous pouvez spécifier si vous souhaitez qu'Amazon S3 chiffre vos données avec des clés gérées par Amazon S3 (SSE-S3) en ajoutant l'en-tête `x-amz-server-side-encryption` à la demande. Définissez la valeur de l'en-tête sur l'algorithme de chiffrement `AES256` pris en charge par Amazon S3. Amazon S3 confirme que votre objet est stocké avec SSE-KMS en renvoyant l'en-tête de réponse `x-amz-server-side-encryption`.

Les opérations d'API de chargement REST suivantes acceptent l'en-tête de demande `x-amz-server-side-encryption`.

- PUT Object
- PUT Object - Copy
- POST Object
- Lancement du chargement partitionné

Lors du chargement d'objets volumineux à l'aide de l'opération d'API de chargement partitionné, vous pouvez spécifier un chiffrement côté serveur en ajoutant l'en-tête `x-amz-server-side-encryption` à la demande de lancement de chargement partitionné. Lorsque vous copiez un objet existant, que l'objet source soit chiffré ou non, l'objet de destination n'est pas chiffré sauf si vous demandez explicitement un chiffrement côté serveur.

Les en-têtes de réponse des opérations d'API REST suivantes renvoient l'en-tête `x-amz-server-side-encryption` lorsqu'un objet est stocké grâce à SSE-S3.

- PUT Object
- PUT Object - Copy
- POST Object
- Lancement du chargement partitionné
- Chargement d'une partie
- Chargement d'une partie (Copy)
- Achèvement du chargement partitionné
- Get Object
- Head Object

> **Note**
>
> N'envoyez pas d'en-têtes de demande de chiffrement pour les demandes `GET` et `HEAD` si votre objet utilise SSE-S3 ou vous obtiendrez une erreur code d'état HTTP 400 (Demande erronée).

Utilisation des AWS SDK

Lorsque vous utilisez AWS des kits SDK, vous pouvez demander à Amazon S3 d'utiliser le chiffrement côté serveur avec des clés de chiffrement gérées par Amazon S3 (SSE-S3). Cette

section fournit des exemples d'utilisation des AWS SDK dans plusieurs langues. Pour plus d'informations sur les autres kits SDK, consultez Exemples de code et de bibliothèques.

Java

Lorsque vous utilisez le AWS SDK for Java pour télécharger un objet, vous pouvez utiliser SSE-S3 pour le chiffrer. Pour demander un chiffrement côté serveur, utilisez la propriété `ObjectMetadata` de la demande `PutObjectRequest` pour définir l'en-tête de demande `x-amz-server-side-encryption`. Lorsque vous appelez la méthode `putObject()` du `AmazonS3Client`, Amazon S3 chiffre et enregistre les données.

Vous pouvez également demander le chiffrement SSE-S3 lors du chargement d'objet avec l'opération d'API de chargement partitionné :

- Lorsque vous utilisez l'opération d'API de chargement partitionné de haut niveau, vous utilisez les méthodes `TransferManager` pour appliquer le chiffrement côté serveur aux objets à mesure que vous les chargez. Vous pouvez utiliser n'importe quelle méthode de chargement qui accepte `ObjectMetadata` comme paramètre. Pour plus d'informations, consultez Chargement d'un objet à l'aide du chargement partitionné.
- Lorsque vous utilisez l'opération d'API de chargement partitionné de bas niveau, vous spécifiez le chiffrement côté serveur quand vous lancez le chargement partitionné. Vous ajoutez la propriété `ObjectMetadata` en appelant la méthode `InitiateMultipartUploadRequest.setObjectMetadata()`. Pour plus d'informations, consultez Utilisation des AWS SDK (API de bas niveau).

Vous ne pouvez pas modifier directement l'état de chiffrement d'un objet (chiffrer un objet non chiffré ou déchiffrer un objet chiffré). Pour modifier l'état de chiffrement d'un objet, vous effectuez une copie de l'objet, en spécifiant l'état de chiffrement voulu pour la copie, puis supprimez l'objet d'origine. Amazon S3 chiffre l'objet copié uniquement si vous demandez explicitement un chiffrement côté serveur. Pour demander le chiffrement de l'objet copié via l'API Java, utilisez la propriété `ObjectMetadata` pour spécifier le chiffrement côté serveur dans la demande `CopyObjectRequest`.

Example Exemple

L'exemple suivant illustre comment définir le chiffrement côté serveur à l'aide du kit AWS SDK for Java. Il explique comment effectuer les tâches suivantes :

- Chargez un nouvel objet à l'aide de SSE-S3.

- Modifier l'état de chiffrement d'un objet (dans cet exemple, chiffrer un objet précédemment non chiffré) en effectuant une copie de l'objet.
- Vérifier l'état de chiffrement de l'objet.

Pour plus d'informations sur le chiffrement côté serveur, consultez Utilisation de l'API REST. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.internal.SSEResultBase;
import com.amazonaws.services.s3.model.*;

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyNameToEncrypt = "*** Key name for an object to upload and encrypt
 ***";
        String keyNameToCopyAndEncrypt = "*** Key name for an unencrypted object to
 be encrypted by copying ***";
        String copiedObjectKeyName = "*** Key name for the encrypted copy of the
 unencrypted object ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Upload an object and encrypt it with SSE.
            uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

            // Upload a new unencrypted object, then change its encryption state
```

```
            // to encrypted by making a copy.
            changeSSEEncryptionStatusByCopying(s3Client,
                    bucketName,
                    keyNameToCopyAndEncrypt,
                    copiedObjectKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
        String objectContent = "Test object encrypted with SSE";
        byte[] objectBytes = objectContent.getBytes();

        // Specify server-side encryption.
        ObjectMetadata objectMetadata = new ObjectMetadata();
        objectMetadata.setContentLength(objectBytes.length);

objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        PutObjectRequest putRequest = new PutObjectRequest(bucketName,
                keyName,
                new ByteArrayInputStream(objectBytes),
                objectMetadata);

        // Upload the object and check its encryption status.
        PutObjectResult putResult = s3Client.putObject(putRequest);
        System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
        printEncryptionStatus(putResult);
    }

    private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
            String bucketName,
            String sourceKey,
            String destKey) {
        // Upload a new, unencrypted object.
        PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey,
"Object example to encrypt by copying");
```

```
        System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
        printEncryptionStatus(putResult);

        // Make a copy of the object and use server-side encryption when storing the
        // copy.
        CopyObjectRequest request = new CopyObjectRequest(bucketName,
                sourceKey,
                bucketName,
                destKey);
        ObjectMetadata objectMetadata = new ObjectMetadata();

 objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        request.setNewObjectMetadata(objectMetadata);

        // Perform the copy operation and display the copy's encryption status.
        CopyObjectResult response = s3Client.copyObject(request);
        System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
        printEncryptionStatus(response);

        // Delete the original, unencrypted object, leaving only the encrypted copy
 in
        // Amazon S3.
        s3Client.deleteObject(bucketName, sourceKey);
        System.out.println("Unencrypted object \"" + sourceKey + "\" deleted.");
    }

    private static void printEncryptionStatus(SSEResultBase response) {
        String encryptionStatus = response.getSSEAlgorithm();
        if (encryptionStatus == null) {
            encryptionStatus = "Not encrypted with SSE";
        }
        System.out.println("Object encryption status is: " + encryptionStatus);
    }
}
```

.NET

Lorsque vous chargez un objet, vous pouvez indiquer à Amazon S3 de le chiffrer. Pour modifier l'état de chiffrement d'un objet existant, vous effectuez une copie de l'objet et supprimez l'objet source. Par défaut, l'opération de copie ne chiffre la cible que si vous demandez explicitement un chiffrement côté serveur de l'objet cible. Pour spécifier SSE-S3 dans `CopyObjectRequest`, ajoutez ce qui suit :

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Pour obtenir un exemple pratique sur la façon de copier un objet, consultez Utilisation des AWS SDK.

L'exemple suivant permet de charger un objet. Dans la demande, l'exemple indique à Amazon S3 de chiffrer l'objet. L'exemple récupère ensuite les métadonnées de l'objet et vérifie la méthode de chiffrement utilisée. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for object created ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
```

```
                    ContentBody = "sample text",
                    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
                };

                var putResponse = await client.PutObjectAsync(putRequest);

                // Determine the encryption state of an object.
                GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };
                GetObjectMetadataResponse response = await
 client.GetObjectMetadataAsync(metadataRequest);
                ServerSideEncryptionMethod objectEncryption =
 response.ServerSideEncryptionMethod;

                Console.WriteLine("Encryption method used: {0}",
 objectEncryption.ToString());
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
 an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Cette rubrique explique comment utiliser les classes de la version 3 de AWS SDK for PHP pour ajouter SSE-S3 aux objets que vous chargez sur Amazon S3. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Pour charger un objet dans Amazon S3, utilisez la méthode Aws\S3\S3Client::putObject(). Pour ajouter l'en-tête de demande `x-amz-server-side-encryption` à votre demande de

chargement, spécifiez le paramètre `ServerSideEncryption` avec la valeur `AES256` comme illustré dans l'exemple de code suivant. Pour plus d'informations sur les demandes de chiffrement côté serveur, consultez Utilisation de l'API REST.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Upload a file with server-side encryption.
$result = $s3->putObject([
    'Bucket'               => $bucket,
    'Key'                  => $keyname,
    'SourceFile'           => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

En réponse, Amazon S3 renvoie l'en-tête `x-amz-server-side-encryption` avec la valeur de l'algorithme de chiffrement qui a été utilisé pour chiffrer les données de l'objet.

Lorsque vous chargez des objets volumineux à l'aide de l'opération d'API de chargement partitionné, vous pouvez spécifier SSE-S3 pour les objets que vous chargez, comme suit :

- Lorsque vous utilisez l'opération d'API de téléchargement partitionné de bas niveau, spécifiez le chiffrement côté serveur lorsque vous appelez la méthode Aws \ S3 \ S3Client : : (). createMultipartUpload Pour ajouter l'en-tête `x-amz-server-side-encryption` à la demande, spécifiez le paramètre `array` avec la clé `ServerSideEncryption` en lui donnant la valeur `AES256`. Pour plus d'informations sur l'opération d'API de chargement partitionné de bas niveau, consultez Utilisation des AWS SDK (API de bas niveau).

- Lorsque vous utilisez l'opération d'API de téléchargement partitionné de haut niveau, spécifiez le chiffrement côté serveur en utilisant le `ServerSideEncryption` paramètre de l'CreateMultipartUploadopération d'API. Pour obtenir un exemple d'utilisation de la méthode `setOption()` avec l'opération d'API de chargement partitionné de haut niveau, consultez Chargement d'un objet à l'aide du chargement partitionné.

Pour déterminer l'état de chiffrement d'un objet existant, récupérez les métadonnées d'objet en appelant la méthode Aws\S3\S3Client::headObject() comme illustré dans l'exemple de code PHP suivant.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Check which server-side encryption algorithm is used.
$result = $s3->headObject([
    'Bucket' => $bucket,
    'Key'    => $keyname,
]);
echo $result['ServerSideEncryption'];
```

Pour changer l'état de chiffrement d'un objet existant, faites une copie de l'objet grâce à la méthode Aws\S3\S3Client::copyObject() et supprimez l'objet source. Par défaut, `copyObject()` ne chiffre pas la cible, sauf si vous demandez explicitement un chiffrement côté serveur de l'objet de destination à l'aide du paramètre `ServerSideEncryption` avec la valeur `AES256`. L'exemple de code PHP suivant fait une copie d'un objet et ajoute un chiffrement côté serveur à l'objet copié.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;
```

```
$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';

$targetBucket = '*** Your Target Bucket Name ***';
$targetKeyname = '*** Your Target Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object and add server-side encryption.
$s3->copyObject([
    'Bucket'               => $targetBucket,
    'Key'                  => $targetKeyname,
    'CopySource'           => "$sourceBucket/$sourceKeyname",
    'ServerSideEncryption' => 'AES256',
]);
```

Pour plus d'informations, consultez les rubriques suivantes :

- AWS SDK for PHP pour la classe Amazon S3 Aws \ S3 \ S3Client
- Documentation AWS SDK for PHP

Ruby

Lorsque vous utilisez le AWS SDK for Ruby pour télécharger un objet, vous pouvez spécifier que l'objet doit être stocké chiffré au repos avec SSE-S3. Lorsque vous relisez l'objet, il est automatiquement déchiffré.

L'exemple de AWS SDK for Ruby version 3 suivant montre comment spécifier qu'un fichier chargé sur Amazon S3 soit chiffré au repos.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
```

```
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
 object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

L'exemple de code suivant montre comment déterminer l'état de chiffrement d'un objet existant.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end
```

```
  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def get_object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
 obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Si un chiffrement côté serveur n'est pas utilisé pour l'objet stocké dans Amazon S3, la méthode renvoie `null`.

Pour modifier l'état de chiffrement d'un objet existant, effectuez une copie de l'objet et supprimez l'objet source. Par défaut, les méthodes de copie ne chiffrent la cible que si vous demandez explicitement un chiffrement côté serveur. Vous pouvez demander le chiffrement de l'objet cible en spécifiant la valeur `server_side_encryption` dans l'argument de hachage d'options, comme illustré dans l'exemple de code Ruby suivant. L'exemple de code montre comment copier un objet et chiffrer la copie avec SSE-S3.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
```

```
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
 key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
 server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key} and "\
```

```
      "encrypted the target with #{target_object.server_side_encryption}
 encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

À l'aide du AWS CLI

Pour spécifier SSE-S3 lorsque vous téléchargez un objet à l'aide du AWS CLI, utilisez l'exemple suivant.

```
aws s3api put-object --bucket example-s3-bucket1 --key object-key-name --server-side-
encryption AES256  --body file path
```

Pour plus d'informations, consultez la section put-object dans la Référence de la AWS CLI . Pour spécifier SSE-S3 lorsque vous copiez un objet à l'aide du AWS CLI, voir copy-object.

En utilisant AWS CloudFormation

Pour des exemples de configuration du chiffrement à l'aide d'un exemple AWS CloudFormation, reportez-vous aux sections Création d'un compartiment avec chiffrement par défaut et Création d'un compartiment en utilisant le chiffrement AWS KMS côté serveur avec une clé de compartiment S3 dans la `AWS::S3::Bucket ServerSideEncryptionRule` rubrique du Guide de l'AWS CloudFormation utilisateur.

Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Le chiffrement côté serveur est le chiffrement des données à leur destination par l'application ou le service qui les reçoit.

Amazon S3 active automatiquement le chiffrement côté serveur avec les clés gérées par Amazon SS3 (SSE-S3) pour les nouveaux chargements d'objets.

Sauf indication contraire, les compartiments utilisent SSE-S3 par défaut pour chiffrer les objets. Toutefois, vous pouvez choisir de configurer les buckets pour qu'ils utilisent plutôt le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS). Pour plus d'informations, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS).

AWS KMS est un service qui combine du matériel et des logiciels sécurisés et hautement disponibles pour fournir un système de gestion des clés adapté au cloud. Amazon S3 utilise le chiffrement côté serveur avec AWS KMS (SSE-KMS) pour chiffrer les données de vos objets S3. De plus, lorsque le SSE-KMS est demandé pour l'objet, la somme de contrôle S3 (dans le cadre des métadonnées de l'objet) est stockée sous forme cryptée. Pour plus d'informations sur le total de contrôle, consultez Vérification de l'intégrité des objets.

Si vous utilisez des clés KMS, vous pouvez les utiliser AWS KMS via l'API AWS Management Consoleou l'AWS KMS API pour effectuer les opérations suivantes :

- Créez, visualisez, modifiez, surveillez, activez ou désactivez, faites tourner et planifiez la suppression des clés KMS de manière centralisée.
- Définir les politiques qui contrôlent comment et par qui les clés KMS peuvent être utilisées.
- Auditer leur utilisation pour prouver qu'elles sont utilisées correctement. L'audit est pris en charge par l'API AWS KMS, mais pas par l'application AWS KMSAWS Management Console.

Les contrôles de sécurité intégrés AWS KMS peuvent vous aider à respecter les exigences de conformité liées au chiffrement. Vous pouvez utiliser ces clés KMS pour protéger les données dans les compartiments Simple Storage Service (Amazon S3). Lorsque vous utilisez le chiffrement SSE-KMS avec un compartiment S3, celui-ci AWS KMS keys doit se trouver dans la même région que le compartiment.

L'utilisation entraîne des frais supplémentaires AWS KMS keys. Pour plus d'informations, consultez la section Concepts AWS KMS key dans le Guide du développeur AWS Key Management Service et Tarification AWS KMS.

Autorisations

Pour télécharger un objet chiffré avec un AWS KMS key vers Amazon S3, vous devez `kms:GenerateDataKey` disposer d'autorisations sur la clé. Pour télécharger un objet chiffré avec un AWS KMS key, vous devez `kms:Decrypt` disposer d'autorisations. Pour plus d'informations sur les AWS KMS autorisations requises pour les téléchargements partitionnés, consultez. API de chargement partitionné et autorisations

> **Important**
>
> Examinez attentivement les autorisations accordées dans vos politiques clés KMS. Limitez toujours les autorisations de politique clé KMS gérées par le client uniquement aux principaux et AWS services IAM qui doivent accéder à l'action clé appropriée. AWS KMS Pour plus d'informations, consultez la section Politiques clés dans AWS KMS.

Rubriques

- AWS KMS keys
- Clés de compartiment Amazon S3
- Exigence du chiffrement côté serveur
- Contexte de chiffrement
- Envoi de demandes pour des objets AWS KMS chiffrés
- Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS)
- Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3

AWS KMS keys

Lorsque vous utilisez le chiffrement côté serveur avec AWS KMS (SSE-KMS), vous pouvez utiliser la cléAWS gérée par défaut ou vous pouvez spécifier une clé gérée par le client que vous avez déjà créée. AWS KMS prend en charge le chiffrement des enveloppes. S3 utilise les AWS KMS fonctionnalités de chiffrement des enveloppes pour mieux protéger vos données. Le chiffrement d'enveloppe est la pratique consistant à chiffrer vos données en texte brut à l'aide d'une clé de données, puis à chiffrer cette clé de données avec une clé KMS. Pour plus d'informations sur le chiffrement d'enveloppe, consultez Chiffrement d'enveloppe dans le Guide du développeur AWS Key Management Service .

Si vous ne spécifiez pas de clé gérée par le client, Amazon S3 Clé gérée par AWS en crée automatiquement une lorsque vous Compte AWS ajoutez un objet chiffré avec SSE-KMS à un bucket pour la première fois. Par défaut, Amazon S3 utilise cette clé KMS pour SSE-KMS.

> **Note**
>
> Les objets chiffrés en utilisant SSE-KMS avec des Clés gérées par AWS ne peuvent pas être partagés entre comptes. Si vous devez partager des données SSE-KMS entre comptes, vous devez utiliser une clé gérée par le client provenant de. AWS KMS

Si vous souhaitez utiliser une clé gérée par le client pour SSE-KMS, créez une clé gérée par le client à chiffrement symétrique avant de configurer SSE-KMS. Ensuite, lorsque vous configurez SSE-KMS pour votre compartiment, vous pouvez spécifier la clé gérée par le client existante. Pour plus d'informations sur la clé de chiffrement symétrique, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

En créant une clé gérée par le client, vous disposez de plus de flexibilité et d'un contrôle accru. Par exemple, vous pouvez créer, faire tourner et désactiver les clés gérés par le client. Vous pouvez également définir des contrôles d'accès et auditer les clés gérées par le client que vous utilisez pour protéger vos données. Pour plus d'informations sur les clés gérées et AWS gérées par le client, consultez la section Clés et AWS clés client dans le guide du AWS Key Management Service développeur.

> **Note**
>
> Lorsque vous utilisez le chiffrement côté serveur avec une clé gérée par le client qui est stockée dans un magasin de clés externe, contrairement aux clés KMS standard, vous êtes responsable de la disponibilité et de la durabilité de votre matériel de clé. Pour plus d'informations sur les magasins de clés externes et sur la manière dont ils modifient le modèle de responsabilité partagée, consultez la section External key stores (Magasins de clés externes) du Guide du développeur AWS Key Management Service .

Utilisation du chiffrement SSE-KMS pour les opérations intercomptes

Tenez compte des éléments suivants lors de l'utilisation du chiffrement pour les opérations inter-comptes :

- Si aucun nom de ressource AWS KMS key Amazon (ARN) ou alias n'est fourni au moment de la demande ou via la configuration de chiffrement par défaut du bucket, le Clé gérée par AWS (`aws/s3`) est utilisé.
- Si vous téléchargez ou accédez à des objets S3 à l'aide de principes AWS Identity and Access Management (IAM) identiques Compte AWS à ceux de votre clé KMS, vous pouvez utiliser le Clé gérée par AWS (). `aws/s3`
- Utilisez une clé gérée par le client si vous souhaitez accorder un accès intercompte à vos objets S3. Vous pouvez configurer la politique d'une clé gérée par le client afin d'autoriser l'accès à partir d'un autre compte.
- Si vous spécifiez une clé KMS gérée par le client, nous vous recommandons d'utiliser un ARN de clé KMS entièrement qualifié. Si vous utilisez plutôt un alias de clé KMS, AWS KMS la clé est résolue dans le compte du demandeur. En raison de ce comportement, les données peuvent être chiffrées avec une clé KMS qui appartient au demandeur, et non au propriétaire du compartiment.
- Vous devez spécifier une clé pour laquelle vous (le demandeur) avez obtenu l'autorisation de `Encrypt`. Pour en savoir plus, consultez Permettre aux utilisateurs de clés d'utiliser une clé KMS pour les opérations de chiffrement dans le Guide de l'utilisateur AWS Key Management Service .

Pour plus d'informations sur les circonstances dans lesquelles utiliser des clés gérées par le client et des clés KMS AWS gérées, consultez Dois-je utiliser une clé gérée par le client Clé gérée par AWS ou une clé gérée par le client pour chiffrer mes objets dans Amazon S3 ?

Flux de travail de chiffrement SSE-KMS

Si vous choisissez de chiffrer vos données à l'aide d'une clé gérée par le client Clé gérée par AWS ou d'une clé gérée par le client, AWS KMS et qu'Amazon S3 exécute les actions de chiffrement d'enveloppe suivantes :

1. Simple Storage Service (Amazon S3) demande une clé de données en texte brut et une copie de la clé chiffrée sous la clé KMS spécifiée.
2. AWS KMS génère une clé de données, la chiffre sous la clé KMS et envoie à la fois la clé de données en texte brut et la clé de données chiffrée à Amazon S3.
3. Amazon S3 chiffre les données à l'aide de la clé de données et supprime la clé en texte brut de la mémoire dès que possible après utilisation.
4. Simple Storage Service (Amazon S3) stocke la clé de données chiffrée sous forme de métadonnées avec les données chiffrées.

Lorsque vous demandez que vos données soient déchiffrées, Amazon S3 AWS KMS effectue les actions suivantes :

1. Amazon S3 envoie la clé de données chiffrée à AWS KMS dans une `Decrypt` demande.
2. AWS KMS déchiffre la clé de données chiffrée à l'aide de la même clé KMS et renvoie la clé de données en texte brut à Amazon S3.
3. Amazon S3 déchiffre les données chiffrées, en utilisant la clé des données en texte brut, et supprime la clé des données en texte brut de la mémoire dès que possible.

> **Important**
>
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 ne prend en charge que les clés KMS à chiffrement symétrique. Pour plus d'informations sur ces clés, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

Audit du chiffrement SSE-KMS

Pour identifier les requêtes qui spécifient SSE-KMS, vous pouvez utiliser les métriques All SSE-KMS requests (Toutes les requêtes SSE-KMS) et % all SSE-KMS requests (% de toutes les requêtes SSE-KMS) dans les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Vous pouvez également utiliser le nombre de compartiments activés par SSE-KMS et le pourcentage de compartiments activés par SSE-KMS pour comprendre le nombre de compartiments (SSE-KMS) utilisés pour le chiffrement des compartiments par défaut. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec S3 Storage Lens. Pour obtenir la liste complète des métriques, consultez le Glossaire des métriques S3 Storage Lens.

Pour vérifier l'utilisation de vos AWS KMS clés pour vos données cryptées SSE-KMS, vous pouvez utiliser AWS CloudTrail des journaux. Vous pouvez obtenir un aperçu de vos opérations cryptographiques, telles que GenerateDataKeyet Decrypt. CloudTrail prend en charge de nombreuses valeurs d'attribut pour filtrer votre recherche, notamment le nom de l'événement, le nom d'utilisateur et la source de l'événement.

Clés de compartiment Amazon S3

Lorsque vous configurez le chiffrement côté serveur à l'aide de AWS KMS (SSE-KMS), vous pouvez configurer vos compartiments pour utiliser les clés de compartiment S3 pour SSE-KMS. L'utilisation d'une clé au niveau du compartiment pour SSE-KMS peut réduire les coûts de vos AWS KMS demandes jusqu'à 99 % en diminuant le trafic de demandes d'Amazon S3 vers. AWS KMS

Lorsque vous configurez un compartiment de sorte qu'il utilise des clés de compartiment S3 pour SSE-KMS sur de nouveaux objets, AWS KMS génère une clé de niveau compartiment qui est utilisée pour créer des clés de données uniques pour les objets dans le compartiment. Cette clé de compartiment S3 est utilisée pendant une période limitée dans le temps dans Amazon S3, ce qui réduit encore la nécessité pour Amazon S3 de faire des demandes AWS KMS pour effectuer des opérations de chiffrement. Pour plus d'informations sur l'utilisation des clés de compartiment S3, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Exigence du chiffrement côté serveur

Pour exiger le chiffrement côté serveur de tous les objets d'un compartiment Simple Storage Service (Amazon S3) particulier, vous pouvez utiliser une politique de compartiment. Par exemple, la politique de compartiment suivante n'autorise pas le chargement d'objet (`s3:PutObject`) si la demande n'inclut pas l'en-tête `x-amz-server-side-encryption-aws-kms-key-id` demandant le chiffrement côté serveur avec SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::example-s3-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Pour exiger qu'une donnée AWS KMS key soit utilisée pour chiffrer les objets d'un compartiment, vous pouvez utiliser la clé de `s3:x-amz-server-side-encryption-aws-kms-key-id` condition. Pour spécifier la clé KMS, vous devez utiliser une clé Amazon Resource Name (ARN) au `arn:aws:kms:region:acct-id:key/key-id` format suivant. AWS Identity and Access Management ne valide pas si la chaîne pour `s3:x-amz-server-side-encryption-aws-kms-key-id` existe.

> **Note**
>
> Lorsque vous chargez un objet, vous pouvez spécifier la clé KMS à l'aide de l'en-tête `x-amz-server-side-encryption-aws-kms-key-id`. Si l'en-tête n'est pas présent dans la demande, Amazon S3 suppose que vous souhaitez utiliser la Clé gérée par AWS. Quoi qu'il en soit, l'ID de AWS KMS clé utilisé par Amazon S3 pour le chiffrement des objets doit correspondre à l'ID de AWS KMS clé indiqué dans la politique, sinon Amazon S3 refuse la demande.

Pour obtenir la liste complète des clés de condition spécifiques à Amazon S3, consultez la section Clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Contexte de chiffrement

Un contexte de chiffrement est un ensemble de paires valeur-clé qui contient des informations contextuelles supplémentaires sur les données. Le contexte de chiffrement n'est pas chiffré. Lorsqu'un contexte de chiffrement est spécifié pour une opération de chiffrement, Amazon S3 doit spécifier le même contexte de chiffrement pour l'opération de déchiffrement. Dans le cas contraire, le déchiffrement échoue. AWS KMS utilise le contexte de chiffrement en tant que données authentifiées supplémentaires (AAD) pour prendre en charge le chiffrement authentifié. Pour plus d'informations sur le contexte de chiffrement, consultez la section Contexte de chiffrement du Guide du développeur AWS Key Management Service .

Par défaut, Amazon S3 utilise l'Amazon Resource Name (ARN) de l'objet ou du compartiment comme paire de contexte de chiffrement :

- Si vous utilisez SSE-KMS sans activer une clé de compartiment S3, l'ARN de l'objet est utilisé comme contexte de chiffrement.

```
arn:aws:s3:::object_ARN
```

- Si vous utilisez SSE-KMS avec une clé de compartiment S3 activée, l'ARN du compartiment est utilisé comme contexte de chiffrement. Pour plus d'informations sur les clés de compartiment S3, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

```
arn:aws:s3:::bucket_ARN
```

Vous pouvez éventuellement fournir une paire de contextes de chiffrement supplémentaire en utilisant l'`x-amz-server-side-encryption-context`en-tête dans une PutObject requête s3 :. Toutefois, étant donné que le contexte de chiffrement n'est pas chiffré, assurez-vous qu'il n'inclut pas d'informations sensibles. Amazon S3 stocke cette paire de clés supplémentaire avec le contexte de chiffrement par défaut. Lorsqu'il traite votre demande PUT, Amazon S3 ajoute le contexte de chiffrement par défaut d'`aws:s3:arn` à celui que vous fournissez.

Vous pouvez utiliser le contexte de chiffrement pour identifier et classer vos opérations cryptographiques par catégorie. Vous pouvez également utiliser la valeur ARN du contexte de chiffrement par défaut pour suivre les demandes pertinentes en AWS CloudTrail visualisant quel ARN Amazon S3 a été utilisé avec quelle clé de chiffrement.

Dans le `requestParameters` champ d'un fichier CloudTrail journal, le contexte de chiffrement est similaire au suivant.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::example-s3-bucket1/file_name"
}
```

Lorsque vous utilisez SSE-KMS avec la fonction de clés de compartiment S3 facultative, la valeur du contexte de chiffrement est l'ARN du compartiment.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::example-s3-bucket1"
}
```

Envoi de demandes pour des objets AWS KMS chiffrés

> **Important**
>
> Toutes GET les PUT demandes d'objets AWS KMS chiffrés doivent être effectuées à l'aide du protocole SSL (Secure Sockets Layer) ou du protocole TLS (Transport Layer Security).

Les demandes doivent également être signées à l'aide d'informations d'identification valides, telles que AWS Signature Version 4 (ou AWS Signature Version 2).

AWS Signature Version 4 est le processus d'ajout d'informations d'authentification aux AWS demandes envoyées par HTTP. Pour des raisons de sécurité, la plupart des demandes AWS doivent être signées avec une clé d'accès, qui consiste en un identifiant de clé d'accès et une clé d'accès secrète. Ces deux clés sont généralement appelées informations d'identification de sécurité. Pour plus d'informations, consultez Authentification des demandes (AWS Signature Version 4) et Processus de signature Signature version 4.

Important

Si votre objet utilise SSE-KMS, n'envoyez pas d'en-têtes de chiffrement pour les requêtes `GET` et `HEAD`. Sinon, vous obtiendrez une erreur HTTP 400 Bad Request (HTTP 400 Requête erronée).

Rubriques

- Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS)
- Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3

Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS)

Important

Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

Vous pouvez appliquer le chiffrement lorsque vous chargez un nouvel objet ou copiez un objet existant.

Vous pouvez spécifier SSE-KMS à l'aide de la console Amazon S3, des opérations d'API REST, des AWS SDK et du AWS Command Line Interface ().AWS CLI Pour plus d’informations, consultez les rubriques suivantes.

> **Note**
>
> Vous pouvez utiliser plusieurs régions AWS KMS keys dans Amazon S3. Cependant, Amazon S3 traite actuellement les clés multi-régions comme s'il s'agissait de clés à région unique et n'utilise pas les fonctions multi-régions de la clé. Pour en savoir plus, consultez la section Utilisation des clés multi-régions dans le Guide du développeur AWS Key Management Service .

> **Note**
>
> Si vous souhaitez utiliser une clé KMS appartenant à un autre compte, vous devez être autorisé à utiliser la clé. Pour plus d'informations sur les autorisations intercomptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service .

Utilisation de la console S3

Cette rubrique explique comment définir ou modifier le type de chiffrement d'un objet pour utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) à l'aide de la console Amazon S3.

> **Note**
>
> - Si vous modifiez le chiffrement d'un objet, un nouvel objet est créé pour remplacer l'ancien. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Le rôle qui modifie la propriété devient également le propriétaire du nouvel objet ou (version de l'objet).
> - Si vous modifiez le type de chiffrement d'un objet doté de balises définies par l'utilisateur, vous devez disposer de cette `s3:GetObjectTagging` autorisation. Si vous modifiez le type de chiffrement d'un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`s3:GetObjectTagging`autorisation.
>
>   Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, le type de chiffrement de l'objet sera mis à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

Pour ajouter ou modifier le chiffrement d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
4. Dans la liste Objets, choisissez le nom de l'objet pour lequel vous souhaitez ajouter ou modifier le chiffrement.

   La page de détails de l'objet apparaît, avec plusieurs sections qui affichent les propriétés de votre objet.
5. Choisissez l’onglet Propriétés.
6. Faites défiler la page jusqu'à la section Paramètres de chiffrement côté serveur et choisissez Modifier.

La page Modifier le chiffrement côté serveur s'ouvre.

7. Sous Chiffrement côté serveur, pour Paramètres de chiffrement, choisissez Ignorer les paramètres de chiffrement par défaut du compartiment.
8. Sous Type de chiffrement, choisissez Chiffrement côté serveur avec AWS Key Management Service clés (SSE-KMS).

> **Important**
>
> Si vous utilisez l'option SSE-KMS pour votre configuration de chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les quotas de AWS KMS et sur la procédure à suivre pour demander une augmentation des quotas, consultez Quotas dans le Guide du développeur AWS Key Management Service .

9. Sous CléAWS KMS , choisissez votre clé KMS avec l'une des options suivantes :

   - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis sélectionnez votre Clé KMS dans la liste des clés disponibles.

     La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .
   - Pour saisir l'ARN de la clé KMS, choisissez Enter AWS KMS key ARN, puis entrez l'ARN de votre clé KMS dans le champ qui apparaît.
   - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

     Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous ne pouvez utiliser que les clés KMS disponibles dans le même compartiment Région AWS que le bucket. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous

souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé KMS, puis saisir l'ARN de la clé KMS.
Amazon S3 prend en charge seulement les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

10. Sélectionnez Save Changes (Enregistrer les modifications).

> **Note**
>
> Cette action applique le chiffrement à tous les objets spécifiés. Lorsque vous chiffrez des dossiers, attendez la fin de l'opération d'enregistrement pour ajouter de nouveaux objets au dossier.

Utilisation de l'API REST

Lorsque vous créez un objet, à savoir, lorsque vous chargez un nouvel objet ou copiez un objet existant, vous pouvez spécifier l'utilisation du chiffrement côté serveur avec des AWS KMS keys (SSE-KMS) pour chiffrer vos données. Pour ce faire, ajoutez l'en-tête `x-amz-server-side-encryption` à la demande. Configurez la valeur de l'en-tête sur l'algorithme de chiffrement `aws:kms`. Amazon S3 confirme que l'objet est stocké grâce à SSE-KMS en renvoyant l'en-tête de réponse `x-amz-server-side-encryption`.

Si vous spécifiez l'en-tête `x-amz-server-side-encryption` avec une valeur de `aws:kms`, vous pouvez également utiliser les en-têtes de demandes suivants :

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Rubriques

- Opérations d'API REST Amazon S3 prenant en charge SSE-KMS
- Contexte de chiffrement (x-amz-server-side-encryption-context)

- AWS KMS ID de clé (x-amz-server-side-encryption-aws-kms-key-id)
- Clés de compartiment S3 (x-amz-server-side-encryption-aws-bucket-key-enabled)

Opérations d'API REST Amazon S3 prenant en charge SSE-KMS

Les opérations d'API REST suivantes acceptent les en-têtes de demande `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` et `x-amz-server-side-encryption-context`.

- PutObject – Lorsque vous chargez des données avec l'opération d'API PUT, vous pouvez spécifier ces en-têtes de demande.
- CopyObject – Lorsque vous copiez un objet, vous disposez d'un objet source et d'un objet cible. Lorsque vous transmettez des en-têtes SSE-KMS avec l'`CopyObject`opération, ils ne sont appliqués qu'à l'objet cible. Lorsque vous copiez un objet existant, que l'objet source soit chiffré ou non, l'objet de destination n'est pas chiffré, sauf si vous demandez explicitement le chiffrement côté serveur.
- POST Object— Lorsque vous utilisez une `POST` opération pour télécharger un objet, au lieu des en-têtes de demande, vous fournissez les mêmes informations dans les champs du formulaire.
- CreateMultipartUpload— Lorsque vous chargez des objets volumineux à l'aide de l'opération d'API de téléchargement en plusieurs parties, vous pouvez spécifier ces en-têtes. Vous spécifiez ces en-têtes dans la `CreateMultipartUpload` demande.

Les en-têtes de réponse des opréations d'API REST suivantes renvoient l'en-tête `x-amz-server-side-encryption` lorsqu'un objet est stocké grâce au chiffrement côté serveur.

- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Toutes `GET` les `PUT` demandes relatives à un objet protégé par AWS KMS échouent si vous ne les faites pas à l'aide du protocole SSL (Secure Sockets Layer), du protocole TLS (Transport Layer Security) ou de la version 4 de signature.
> - Si votre objet utilise le SSE-KMS, n'envoyez pas d'en-têtes de demande de chiffrement pour les `GET` requêtes et les `HEAD` requêtes, sinon vous obtiendrez une erreur HTTP 400. BadRequest

Contexte de chiffrement (**`x-amz-server-side-encryption-context`**)

Si vous spécifiez `x-amz-server-side-encryption:aws:kms`, l'API Simple Storage Service (Amazon S3) prend en charge un contexte de chiffrement avec l'en-tête `x-amz-server-side-encryption-context`. Un contexte de chiffrement est un ensemble de paires valeur clé qui contient des informations contextuelles supplémentaires sur les données.

Amazon S3 utilise automatiquement l'Amazon Resource Name (ARN) de l'objet ou du compartiment comme paire de contexte de chiffrement. Si vous utilisez SSE-KMS sans activer une clé de compartiment S3, vous utilisez l'ARN de l'objet comme contexte de chiffrement, par exemple, `arn:aws:s3:::`*`object_ARN`*. Toutefois, si vous utilisez SSE-KMS et activez une clé de compartiment S3, vous utilisez l'ARN du compartiment pour votre contexte de chiffrement, par exemple, `arn:aws:s3:::`*`bucket_ARN`*.

Vous pouvez éventuellement fournir une paire de contexte de chiffrement supplémentaire à l'aide de l'en-tête `x-amz-server-side-encryption-context`. Toutefois, étant donné que le contexte de chiffrement n'est pas chiffré, assurez-vous qu'il n'inclut pas d'informations sensibles. Amazon S3 stocke cette paire de clés supplémentaire avec le contexte de chiffrement par défaut.

Pour plus d'informations sur le contexte de chiffrement dans Simple Storage Service (Amazon S3), consultez Contexte de chiffrement. Pour des informations générales sur le contexte de chiffrement, consultez la section Concepts AWS Key Management Service - Contexte de chiffrement du Guide du développeur AWS Key Management Service .

AWS KMS ID de clé (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Vous pouvez utiliser l'en-tête `x-amz-server-side-encryption-aws-kms-key-id` pour spécifier l'ID de la clé gérée par le client utilisée pour protéger les données. Si vous spécifiez l'en-tête

`x-amz-server-side-encryption:aws:kms` mais que vous ne fournissez pas l'en-tête `x-amz-server-side-encryption-aws-kms-key-id`, Amazon S3 utilisera la Clé gérée par AWS (`aws/s3`) pour protéger les données. Si vous souhaitez utiliser une clé gérée par le client, vous devrez fournir l'en-tête `x-amz-server-side-encryption-aws-kms-key-id` de la clé gérée par le client.

> **Important**
>
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 ne prend en charge que les clés KMS à chiffrement symétrique. Pour plus d'informations sur ces clés, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

Clés de compartiment S3 (**`x-amz-server-side-encryption-aws-bucket-key-enabled`**)

Vous pouvez utiliser l'en-tête de `x-amz-server-side-encryption-aws-bucket-key-enabled` demande pour activer ou désactiver une clé de compartiment S3 au niveau de l'objet. Les clés de compartiment S3 réduisent les coûts de vos AWS KMS demandes en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Si vous spécifiez l'en-tête `x-amz-server-side-encryption:aws:kms`, mais que vous ne fournissez pas l'en-tête `x-amz-server-side-encryption-aws-bucket-key-enabled`, votre objet utilise les paramètres de clé de compartiment S3 du compartiment de destination pour chiffrer votre objet. Pour plus d'informations, consultez Configuration d'une clé de compartiment S3 au niveau d'un objet .

À l'aide du AWS CLI

Pour utiliser les exemples de AWS CLI commandes suivants, remplacez-les *`user input placeholders`* par vos propres informations.

Lorsque vous chargez un nouvel objet ou que vous copiez un objet existant, vous pouvez spécifier l'utilisation du chiffrement côté serveur avec des AWS KMS clés pour chiffrer vos données. Pour ce faire, ajoutez l'en-tête `--server-side-encryption aws:kms` à la demande. Utilisez le `--ssekms-key-id` *`example-key-id`* pour ajouter la AWS KMS clé gérée par le client que vous avez

créée. Si vous spécifiez`--server-side-encryption aws:kms`, mais que vous ne fournissez pas d'identifiant de AWS KMS clé, Amazon S3 utilisera une clé AWS gérée.

```
aws s3api put-object --bucket example-s3-bucket --key example-object-key --server-side-encryption aws:kms --ssekms-key-id example-key-id --ssekms-encryption-context example-encryption-context --body filepath
```

Vous pouvez activer ou désactiver les clés de compartiment S3 sur vos `copy-object` opérations `put-object` ou en ajoutant `--bucket-key-enabled` ou`--no-bucket-key-enabled`. Les clés de compartiment S3 peuvent réduire les coûts de vos AWS KMS demandes en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez la section Réduction du coût du SSE-KMS avec les clés de compartiment S3.

```
aws s3api put-object --bucket example-s3-bucket --key example-object-key --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Vous pouvez copier un objet d'un compartiment source vers un nouveau compartiment et spécifier le chiffrement SSE-KMS.

```
aws s3api copy-object --copy-source example-s3-bucket/example-object-key --
bucket example-s3-bucket2 --key example-object-key --server-side-encryption aws:kms --
sse-kms-key-id example-key-id --ssekms-encryption-context example-encryption-context
```

Utilisation des AWS SDK

Lorsque vous utilisez AWS des SDK, vous pouvez demander à Amazon S3 de les utiliser AWS KMS keys pour le chiffrement côté serveur. Les exemples suivants montrent comment utiliser SSE-KMS avec les AWS SDK pour Java et .NET. Pour plus d'informations sur les autres SDK, consultez la section Exemples de code et bibliothèques du AWS Developer Center.

> **Important**
>
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 ne prend en charge que les clés KMS à chiffrement symétrique. Pour plus d'informations sur ces clés, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

Opération **`CopyObject`**

Lors de la copie d'objets, vous ajoutez les mêmes propriétés de demande (`ServerSideEncryptionMethod` et `ServerSideEncryptionKeyManagementServiceKeyId`) pour demander à Amazon S3 d'utiliser une AWS KMS key. Pour plus d'informations sur la copie d'objets, consultez Copier, déplacer et renommer des objets.

Opération **`PUT`**

Java

Lorsque vous chargez un objet à l'aide du AWS SDK for Java, vous pouvez demander à Amazon S3 d'utiliser un AWS KMS key en ajoutant la `SSEAwsKeyManagementParams` propriété comme indiqué dans la demande suivante :

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

Dans ce cas, Amazon S3 utilise le Clé gérée par AWS (`aws/s3`). Pour plus d'informations, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS). Vous pouvez éventuellement créer une clé KMS de chiffrement symétrique et la spécifier dans la demande, comme illustré dans l'exemple suivant :

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new
 SSEAwsKeyManagementParams(keyID));
```

Pour plus d'informations sur la création de clés gérées par le client, consultez la section Programmation de l' AWS KMS API dans le Guide du AWS Key Management Service développeur.

Pour obtenir des exemples de code utilisables pour charger un objet, consultez les rubriques suivantes. Pour utiliser ces exemples, vous devez mettre à jour les exemples de code et fournir des informations de chiffrement comme illustré dans le fragment de code précédent.

- Pour charger un objet en une seule opération, veuillez consulter Chargement d'objets.
- Pour les téléchargements partitionnés utilisant les opérations de l'API de téléchargement partitionné de haut niveau ou de bas niveau, consultez. Chargement d'un objet à l'aide du chargement partitionné

.NET

Lorsque vous chargez un objet à l'aide du AWS SDK for .NET, vous pouvez demander à Amazon S3 d'utiliser un AWS KMS key en ajoutant la `ServerSideEncryptionMethod` propriété comme indiqué dans la demande suivante :

```
PutObjectRequest putRequest = new PutObjectRequest
 {
     BucketName = example-s3-bucket,
     Key = keyName,
     // other properties
     ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
 };
```

Dans ce cas, Amazon S3 utilise le Clé gérée par AWS. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS). Vous pouvez éventuellement créer votre propre clé de chiffrement symétrique gérée par le client et la spécifier dans la demande, comme illustré dans l'exemple suivant :

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
  BucketName = example-s3-bucket,
  Key = keyName,
  // other properties
  ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
  ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Pour plus d'informations sur la création de clés gérées par le client, consultez la section Programmation de l' AWS KMS API dans le Guide du AWS Key Management Service développeur.

Pour obtenir des exemples de code utilisables pour charger un objet, consultez les rubriques suivantes. Pour utiliser ces exemples, vous devez mettre à jour les exemples de code et fournir des informations de chiffrement comme illustré dans le fragment de code précédent.

- Pour charger un objet en une seule opération, veuillez consulter Chargement d'objets.
- Pour les téléchargements partitionnés utilisant les opérations de l'API de téléchargement partitionné de haut niveau ou de bas niveau, consultez. Chargement d'un objet à l'aide du chargement partitionné

URL présignées

Java

Lorsque vous créez une URL présignée pour un objet chiffré avec un AWS KMS key, vous devez spécifier explicitement la version de signature 4, comme illustré dans l'exemple suivant :

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
        new ProfileCredentialsProvider(), clientConfiguration);
...
```

Pour obtenir un exemple de code, consultez Partage d'objets à l'aide d'URL présignées.

.NET

Lorsque vous créez une URL présignée pour un objet chiffré avec un AWS KMS key, vous devez spécifier explicitement la version de signature 4, comme illustré dans l'exemple suivant :

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Pour obtenir un exemple de code, consultez Partage d'objets à l'aide d'URL présignées.

Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3

Les clés de compartiment Amazon S3 réduisent le coût du chiffrement côté serveur Amazon S3 avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS). L'utilisation d'une clé au niveau du compartiment pour SSE-KMS peut réduire les coûts des AWS KMS demandes jusqu'à 99 % en diminuant le trafic de demandes d'Amazon S3 vers. AWS KMS En quelques clics dans la AWS Management Console et sans modifier vos applications clients, vous pouvez configurer votre compartiment de sorte qu'il utilise une clé de compartiment S3 pour le chiffrement SSE-KMS pour les nouveaux objets.

> **Note**
>
> Les clés de compartiment S3 ne sont pas prises en charge pour le chiffrement double couche côté serveur avec des clés AWS Key Management Service (AWS KMS) (DSSE-KMS).

Clés de compartiment S3 pour SSE-KMS

Les charges de travail qui accèdent à des millions ou des milliards d'objets chiffrés avec SSE-KMS peuvent générer d'importants volumes de demandes à. AWS KMS Lorsque vous utilisez SSE-KMS pour protéger vos données sans clé de compartiment S3, Amazon S3 utilise une clé de AWS KMS données individuelle pour chaque objet. Dans ce cas, Amazon S3 effectue un appel à AWS KMS chaque fois qu'une demande est faite contre un objet chiffré par KMS. Pour plus d'informations sur le fonctionnement de SSE-KMS, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

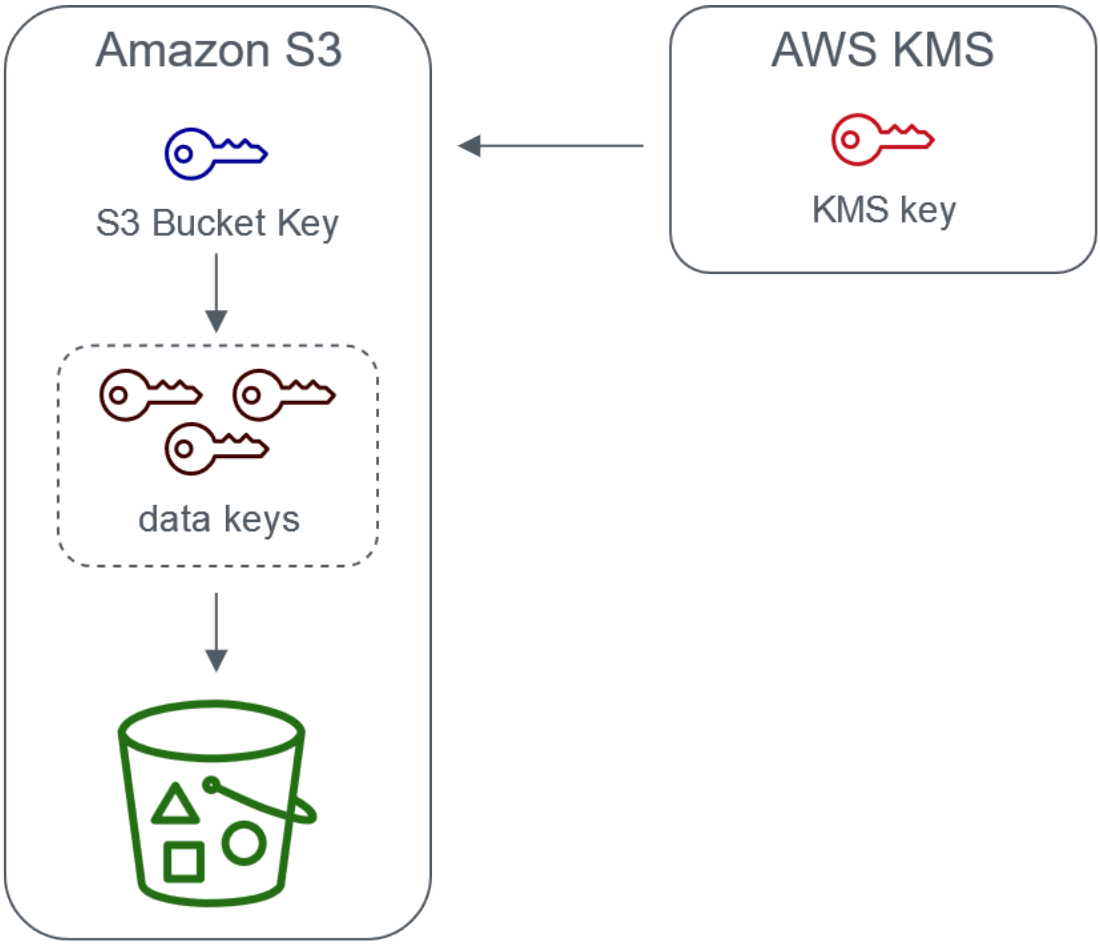
Lorsque vous configurez votre compartiment pour utiliser une clé de compartiment S3 pour SSE-KMS, vous AWS générez une clé de type bucket de courte durée à partir de AWS KMS, puis la conservez temporairement dans S3. Cette clé de niveau compartiment créera des clés de données pour les nouveaux objets au cours de son cycle de vie. Les clés de compartiment S3 sont utilisées pendant une période limitée dans Amazon S3, ce qui réduit la nécessité pour S3 de faire des demandes AWS KMS pour effectuer des opérations de chiffrement. Cela réduit le trafic de S3 à S3 AWS KMS, ce qui vous permet d'accéder à des objets AWS KMS chiffrés dans Amazon S3 à une fraction du coût précédent.

Les clés uniques au niveau du compartiment sont récupérées au moins une fois par demandeur afin de garantir que l'accès du demandeur à la clé est capturé lors d'un événement. AWS KMS CloudTrail Amazon S3 traite les appelants comme des demandeurs différents lorsqu'ils utilisent des rôles ou des comptes différents, ou lorsqu'ils utilisent le même rôle avec des politiques de cadrage différentes. AWS KMS les économies de demandes reflètent le nombre de demandeurs, les modèles de demandes et l'âge relatif des objets demandés. Par exemple, un nombre réduit de demandeurs, sollicitant plusieurs objets dans une fenêtre de temps limitée, et chiffrés avec la même clé au niveau des compartiments, permettra de réaliser des économies plus importantes.

> **Note**
>
> L'utilisation de clés de compartiment S3 vous permet de réduire les coûts liés aux AWS KMS demandes en réduisant le nombre de demandes à AWS KMS for `Encrypt` et les `Decrypt` opérations grâce à l'utilisation d'une clé au niveau du compartiment. `GenerateDataKey` De par leur conception, les demandes ultérieures qui tirent parti de cette clé au niveau du compartiment n'entraînent pas de demandes d' AWS KMS API et ne valident pas l'accès par rapport à la politique de AWS KMS clé.

Lorsque vous configurez une clé de compartiment S3, les objets qui se trouvent déjà dans le compartiment n'utilisent pas la clé de compartiment S3. Pour configurer une clé de compartiment S3 pour des objets existants, vous pouvez utiliser une opération `CopyObject`. Pour plus d'informations, consultez Configuration d'une clé de compartiment S3 au niveau d'un objet .

Amazon S3 partage une clé de compartiment S3 uniquement pour les objets chiffrés avec la même AWS KMS key. Les clés de compartiment S3 sont compatibles avec les clés KMS créées par AWS KMS, le matériel clé importé et le matériel clé soutenu par des magasins de clés personnalisés.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Configuration de clés de compartiment S3

Vous pouvez configurer votre compartiment pour utiliser une clé de compartiment S3 pour SSE-KMS sur de nouveaux objets via la console Amazon S3, les AWS kits SDK ou l'API AWS CLI REST. Lorsque l'option « S3 Bucket Keys » (Clés du compartiment S3) est activée sur votre compartiment, les objets chargés avec une autre clé SSE-KMS spécifiée utiliseront leur propres clés du compartiment S3. Quel que soit votre paramètre de clé de compartiment S3, vous pouvez inclure

l'en-tête `x-amz-server-side-encryption-bucket-key-enabled` avec une valeur `true` ou `false` dans votre requête, afin de remplacer le paramètre de compartiment.

Avant de configurer votre compartiment de sorte qu'il utilise une clé de compartiment S3, consultez Modifications à prendre en compte avant d'activer une clé de compartiment S3.

Configuration d'une clé de compartiment S3 à l'aide de la console Amazon S3

Lorsque vous créez un nouveau compartiment, vous pouvez configurer votre compartiment de sorte qu'il utilise une clé de compartiment S3 pour SSE-KMS sur de nouveaux objets. Vous pouvez également configurer un compartiment existant de sorte qu'il utilise une clé de compartiment S3 pour SSE-KMS sur de nouveaux objets en mettant à jour vos propriétés de compartiment.

Pour plus d'informations, consultez Configuration de votre compartiment de sorte qu'il utilise une clé de compartiment S3 avec SSE-KMS pour de nouveaux objets.

Support de l'API REST et du AWS SDK pour les clés de compartiment S3 AWS CLI

Vous pouvez utiliser l'API REST ou le AWS SDK pour configurer votre compartiment afin qu'il utilise une clé de compartiment S3 pour SSE-KMS sur de nouveaux objets. AWS CLI Vous pouvez également activer une clé de compartiment S3 au niveau de l'objet.

Pour plus d'informations, consultez les ressources suivantes :

- Configuration d'une clé de compartiment S3 au niveau d'un objet
- Configuration de votre compartiment de sorte qu'il utilise une clé de compartiment S3 avec SSE-KMS pour de nouveaux objets

Les opérations d'API suivantes prennent en charge les clés de compartiment S3 pour SSE-KMS :

- PutBucketEncryption
  - `ServerSideEncryptionRule` accepte le `BucketKeyEnabled` paramètre permettant d'activer et de désactiver une clé de compartiment S3.
- GetBucketEncryption
  - `ServerSideEncryptionRule` renvoie les paramètres de `BucketKeyEnabled`.
- PutObject, CopyObjectCreateMultipartUpload, et objet POST
  - L'en-tête de demande `x-amz-server-side-encryption-bucket-key-enabled` active ou désactive une clé de compartiment S3 au niveau de l'objet.
- HeadObject, GetObjectUploadPartCopy, UploadPart, et CompleteMultipartUpload

- L'en-tête de réponse `x-amz-server-side-encryption-bucket-key-enabled` indique si une clé de compartiment S3 est activée ou désactivée pour un objet.

Travailler avec AWS CloudFormation

Dans AWS CloudFormation, la `AWS::S3::Bucket` ressource inclut une propriété de chiffrement appelée `BucketKeyEnabled` que vous pouvez utiliser pour activer ou désactiver une clé de compartiment S3.

Pour plus d'informations, consultez En utilisant AWS CloudFormation.

Modifications à prendre en compte avant d'activer une clé de compartiment S3

Avant d'activer une clé de compartiment S3, notez les modifications suivantes :

IAM ou politiques AWS KMS clés

Si vos politiques AWS Identity and Access Management (IAM) ou AWS KMS clés existantes utilisent votre objet Amazon Resource Name (ARN) comme contexte de chiffrement pour affiner ou limiter l'accès à votre clé KMS, ces politiques ne fonctionneront pas avec une clé de compartiment S3. Les clés de compartiment S3 utilisent l'ARN du compartiment comme contexte de chiffrement. Avant d'activer une clé de compartiment S3, mettez à jour vos politiques IAM ou vos politiques AWS KMS clés pour utiliser l'ARN de votre compartiment comme contexte de chiffrement.

Pour plus d'informations sur le contexte de chiffrement et les clés de compartiment S3, consultez Contexte de chiffrement.

CloudTrail événements pour AWS KMS

Une fois que vous avez activé une clé de compartiment S3, vos AWS KMS CloudTrail événements enregistrent l'ARN de votre compartiment au lieu de l'ARN de votre objet. En outre, vous voyez moins d' CloudTrail événements KMS pour les objets SSE-KMS dans vos journaux. Les informations clés étant limitées dans le temps dans Amazon S3, moins de demandes sont adressées à AWS KMS.

Utilisation d'une clé de compartiment S3 avec réplication

Vous pouvez utiliser des clés de compartiment S3 avec la réplication dans la même Région (SRR) et la réplication entre Régions (CRR).

Lorsqu'Amazon S3 réplique un objet chiffré, il conserve généralement les paramètres de chiffrement de l'objet réplica dans le compartiment de destination. Toutefois, si l'objet source n'est pas chiffré et

que votre compartiment de destination utilise un chiffrement par défaut ou une clé de compartiment S3, Amazon S3 chiffre l'objet avec la configuration du compartiment de destination.

Les exemples suivants illustrent le fonctionnement d'une clé de compartiment S3 avec la réplication. Pour plus d'informations, consultez Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Example Exemple 1 : l'objet source utilise des clés de compartiment S3, le compartiment de destination utilise le chiffrement par défaut

Si votre objet source utilise une clé de compartiment S3 mais que votre compartiment de destination utilise le chiffrement par défaut avec SSE-KMS, l'objet réplica conserve ses paramètres de chiffrement de clé de compartiment S3 dans le compartiment de destination. Le compartiment de destination utilise toujours le chiffrement par défaut avec SSE-KMS.

Example Exemple 2 : l'objet source n'est pas chiffré, le compartiment de destination utilise une clé de compartiment S3 avec SSE-KMS

Si votre objet source n'est pas chiffré et que le compartiment de destination utilise une clé de compartiment S3 avec SSE-KMS, l'objet de réplica est chiffré avec une clé de compartiment S3 utilisant SSE-KMS dans le compartiment de destination. Cela produit un `ETag` de l'objet source différent de l'`ETag` de l'objet réplica. Vous devez mettre à jour les applications qui utilisent le `ETag` pour compenser cette différence.

Utilisation des clés de compartiment S3

Pour plus d'informations sur l'activation et l'utilisation des clés de compartiment S3, consultez les sections suivantes :

- Configuration de votre compartiment de sorte qu'il utilise une clé de compartiment S3 avec SSE-KMS pour de nouveaux objets
- Configuration d'une clé de compartiment S3 au niveau d'un objet
- Affichage des paramètres d'une clé de compartiment S3

Configuration de votre compartiment de sorte qu'il utilise une clé de compartiment S3 avec SSE-KMS pour de nouveaux objets

Lorsque vous configurez le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS), vous pouvez configurer votre compartiment pour utiliser une clé de

compartiment S3 pour SSE-KMS sur de nouveaux objets. Les clés de compartiment S3 réduisent le trafic de requêtes en provenance d'Amazon S3 AWS KMS et réduisent le coût du SSE-KMS. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Vous pouvez configurer votre compartiment pour utiliser une clé de compartiment S3 pour SSE-KMS sur de nouveaux objets à l'aide de la console Amazon S3, de l'API REST, AWS des SDK, AWS Command Line Interface (AWS CLI) ou. AWS CloudFormation Si vous souhaitez activer ou désactiver une clé de compartiment S3 pour des objets existants, vous pouvez utiliser une opération `CopyObject`. Pour plus d'informations, consultez Configuration d'une clé de compartiment S3 au niveau d'un objet et Utilisation d'opérations pat lot S3 pour chiffrer des objets avec des clés de compartiment S3.

Lorsqu'une clé de compartiment S3 est activée pour le compartiment source ou de destination, le contexte de chiffrement est l'Amazon Resource Name (ARN) du compartiment source et non l'ARN de l'objet, par exemple, `arn:aws:s3:::`*`bucket_ARN`*. Vous devez mettre à jour vos stratégies IAM pour utiliser l'ARN du compartiment comme contexte de chiffrement. Pour plus d'informations, consultez Clés de compartiment S3 et réplication.

Les exemples suivants illustrent le fonctionnement d'une clé de compartiment S3 avec la réplication. Pour plus d'informations, consultez Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Prérequis

Avant de configurer votre compartiment de sorte qu'il utilise une clé de compartiment S3, consultez Modifications à prendre en compte avant d'activer une clé de compartiment S3.

Utiliser la console S3.

Dans la console S3, vous pouvez activer ou désactiver une clé de compartiment S3 pour un nouveau compartiment ou un compartiment existant. Les objets de la console S3 conservent le paramètre de clé de compartiment S3 présent dans la configuration du compartiment. Lorsque vous activez une clé de compartiment S3 pour votre compartiment, les nouveaux objets que vous chargez dans le compartiment utilisent une clé de compartiment S3 pour SSE-KMS.

Chargement, copie ou modification d'objets dans des compartiments pour lesquels une clé de compartiment S3 est activée

Si vous chargez, modifiez ou copiez un objet dans un compartiment pour lequel une clé de compartiment S3 est activée, les paramètres de clé de compartiment S3 de cet objet peuvent être mis à jour pour les aligner sur la configuration du compartiment.

Si une clé de compartiment S3 est déjà activée pour un objet, les paramètres de clé de compartiment S3 de cet objet ne changent pas lorsque vous copiez ou modifiez l'objet. Toutefois, si vous modifiez ou copiez un objet pour lequel aucune clé de compartiment S3 n'est activée et que le compartiment de destination a une configuration de clé de compartiment S3, l'objet conserve les paramètres de clé de compartiment S3 du compartiment de destination. Par exemple, si aucune clé de compartiment S3 n'est activée pour votre objet source, mais que la clé de compartiment S3 est activée pour le compartiment de destination, une clé de compartiment S3 est activée pour l'objet.

Pour activer une clé de compartiment S3 lorsque vous créez un nouveau compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.
4. Entrez le nom de votre compartiment, puis choisissez votre Région AWS.
5. Sous Chiffrement par défaut, pour Type de clé de chiffrement, choisissez CléAWS Key Management Service (SSE-KMS).
6. Sous CléAWS KMS , choisissez votre clé KMS avec l'une des options suivantes :

   - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

     La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez la section Clés et AWS clés client dans le Guide du AWS Key Management Service développeur.
   - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
   - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

     Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le guide du AWS Key Management Service développeur.
7. Sous Clé de compartiment, choisissez Activer.

8. Choisissez Créer un compartiment.

   Amazon S3 crée votre compartiment avec une clé de compartiment S3 activée. Les nouveaux objets que vous chargez dans le compartiment utiliseront une clé de compartiment S3.

   Pour désactiver une clé de compartiment S3, suivez les étapes précédentes et choisissez Désactiver.

Pour activer une clé de compartiment S3 pour un compartiment existant

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le compartiment pour lequel vous souhaitez activer une clé de compartiment S3.
4. Choisissez l'onglet Propriétés.
5. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).
6. Sous Chiffrement par défaut, pour Type de clé de chiffrement, choisissez CléAWS Key Management Service (SSE-KMS).
7. Sous CléAWS KMS , choisissez votre clé KMS avec l'une des options suivantes :

   - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

     La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez la section Clés et AWS clés client dans le Guide du AWS Key Management Service développeur.
   - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
   - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

     Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le guide du AWS Key Management Service développeur.
8. Sous Clé de compartiment, choisissez Activer.
9. Sélectionnez Save Changes (Enregistrer les modifications).

Amazon S3 active une clé de compartiment S3 pour les nouveaux objets ajoutés à votre compartiment. Les objets existants n'utilisent pas la clé de compartiment S3. Pour configurer une clé de compartiment S3 pour des objets existants, vous pouvez utiliser une opération `CopyObject`. Pour plus d'informations, consultez Configuration d'une clé de compartiment S3 au niveau d'un objet .

Pour désactiver une clé de compartiment S3, suivez les étapes précédentes et choisissez Désactiver.

Utilisation de l'API REST

Vous pouvez l'utiliser PutBucketEncryptionpour activer ou désactiver une clé de compartiment S3 pour votre compartiment. Pour configurer une clé de compartiment S3 avec`PutBucketEncryption`, utilisez le type de ServerSideEncryptionRuledonnées, qui inclut le chiffrement par défaut avec SSE-KMS. Vous pouvez également utiliser une clé gérée par le client en indiquant l'ID de clé KMS de la clé gérée par le client.

Pour plus d'informations et des exemples de syntaxe, consultez PutBucketEncryption.

Utilisation du AWS SDK pour Java

L'exemple suivant active le chiffrement du compartiment par défaut avec SSE-KMS et une clé de compartiment S3 à l'aide d' AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

ServerSideEncryptionByDefault serverSideEncryptionByDefault = new
 ServerSideEncryptionByDefault()
    .withSSEAlgorithm(SSEAlgorithm.KMS);
ServerSideEncryptionRule rule = new ServerSideEncryptionRule()
    .withApplyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
    .withBucketKeyEnabled(true);
ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
    new ServerSideEncryptionConfiguration().withRules(Collections.singleton(rule));

SetBucketEncryptionRequest setBucketEncryptionRequest = new
 SetBucketEncryptionRequest()
```

```
    .withServerSideEncryptionConfiguration(serverSideEncryptionConfiguration)
    .withBucketName(bucketName);

s3client.setBucketEncryption(setBucketEncryptionRequest);
```

À l'aide du AWS CLI

L'exemple suivant active le chiffrement du compartiment par défaut avec SSE-KMS et une clé de compartiment S3 à l'aide d' AWS CLI. Remplacez *user input placeholders* par vos propres informations.

```
aws s3api put-bucket-encryption --bucket example-s3-bucket --server-side-encryption-
configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

En utilisant AWS CloudFormation

Pour plus d'informations sur la configuration d'une clé de compartiment S3 avec AWS CloudFormation, consultez AWS::S3::Bucket ServerSideEncryptionRulele guide de AWS CloudFormation l'utilisateur.

Configuration d'une clé de compartiment S3 au niveau d'un objet

Lorsque vous effectuez une opération PUT ou COPY à l'aide de l'API AWS REST, des SDK ou AWS CLI, vous pouvez activer ou désactiver une clé de compartiment S3 au niveau de l'objet en ajoutant l'en-tête de `x-amz-server-side-encryption-bucket-key-enabled` demande avec une `false` valeur `true` ou. Les clés de compartiment S3 réduisent le coût du chiffrement côté serveur à l'aide de AWS Key Management Service (AWS KMS) (SSE-KMS) en diminuant le trafic de requêtes d'Amazon S3 vers. AWS KMS Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Lorsque vous configurez une clé de compartiment S3 pour un objet à l'aide d'une opération PUT ou COPY, Amazon S3 met à jour uniquement les paramètres de cet objet. Les paramètres de clé de compartiment S3 pour le compartiment de destination ne changent pas. Si vous soumettez une requête PUT ou COPY pour un objet chiffré par KMS dans un compartiment avec l'option « S3 Bucket Keys » (Clés du compartiment S3) activée, votre opération au niveau de l'objet utilisera automatiquement cette option à moins que vous ne désactiviez les clés dans l'en-tête de la requête. Si vous ne spécifiez pas de clé de compartiment S3 pour votre objet, Amazon S3 applique les paramètres de clé de compartiment S3 du compartiment de destination à l'objet.

Prérequis :

Avant de configurer votre objet de sorte qu'il utilise une clé de compartiment S3, consultez Modifications à prendre en compte avant d'activer une clé de compartiment S3.

Rubriques

- Opérations par lot Amazon S3
- Utilisation de l'API REST
- Utilisation du AWS SDK pour Java PutObject ()
- En utilisant le AWS CLI (PutObject)

Opérations par lot Amazon S3

Pour chiffrer vos objets Amazon S3 existants, vous pouvez utiliser des opérations par lot Amazon S3. Vous fournissez à la fonctionnalité d'opérations par lot S3 une liste d'objets sur lesquels agir. La fonctionnalité d'opérations par lot appelle l'API correspondante pour exécuter l'opération spécifiée.

Vous pouvez utiliser l'opération de copie des opérations par lot S3 pour copier des objets non chiffrés existants et les réécrire dans le même compartiment en tant qu'objets chiffrés. Une tâche d'opérations par lot peut effectuer l'opération spécifiée sur des milliards d'objets. Pour plus d'informations, consultez Exécution des opérations par lot à grande échelle sur des objets Amazon S3 et le billet de blog Encrypting objects with Amazon S3 Batch Operations.

Utilisation de l'API REST

Lorsque vous utilisez SSE-KMS, vous pouvez activer une clé de compartiment S3 pour un objet à l'aide des opérations d'API suivantes :

- PutObject— Lorsque vous chargez un objet, vous pouvez spécifier l'en-tête de `x-amz-server-side-encryption-bucket-key-enabled` demande pour activer ou désactiver une clé de compartiment S3 au niveau de l'objet.
- CopyObject— Lorsque vous copiez un objet et configurez SSE-KMS, vous pouvez spécifier l'en-tête de `x-amz-server-side-encryption-bucket-key-enabled` demande pour activer ou désactiver une clé de compartiment S3 pour votre objet.
- POST Object : lorsque vous utilisez une opération `POST` pour charger un objet et configurer SSE-KMS, vous pouvez utiliser le champ de formulaire `x-amz-server-side-encryption-bucket-key-enabled` pour activer ou désactiver une clé de compartiment S3 pour votre objet.
- CreateMultipartUpload— Lorsque vous chargez des objets volumineux à l'aide de l'opération `CreateMultipartUpload` API et que vous configurez SSE-KMS, vous pouvez utiliser l'en-tête de `x-amz-server-side-encryption-bucket-key-enabled` demande pour activer ou désactiver une clé de compartiment S3 pour votre objet.

Pour activer une clé de compartiment S3 au niveau de l'objet, incluez l'en-tête de demande `x-amz-server-side-encryption-bucket-key-enabled`. Pour plus d'informations sur SSE-KMS et l'API REST, consultez Utilisation de l'API REST.

Utilisation du AWS SDK pour Java PutObject ()

Vous pouvez utiliser l'exemple suivant pour configurer une clé de compartiment S3 au niveau de l'objet à l'aide du kit AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "DOC-EXAMPLE-BUCKET1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
 contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

En utilisant le AWS CLI (PutObject)

Vous pouvez utiliser l' AWS CLI exemple suivant pour configurer une clé de compartiment S3 au niveau de l'objet dans le cadre d'une `PutObject` demande.

```
aws s3api put-object --bucket example-s3-bucket --key object key name --server-side-encryption aws:kms --bucket-key-enabled --body filepath
```

Affichage des paramètres d'une clé de compartiment S3

Vous pouvez consulter les paramètres d'une clé de compartiment S3 au niveau du compartiment ou de l'objet à l'aide de la console Amazon S3, de l'API REST AWS Command Line Interface (AWS CLI) ou AWS des kits SDK.

Les clés de compartiment S3 réduisent le trafic de requêtes en provenance d'Amazon S3 vers Amazon S3 AWS KMS et réduisent le coût du chiffrement côté serveur AWS Key Management Service (SSE-KMS). Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

Pour afficher les paramètres de clé de compartiment S3 d'un compartiment ou d'un objet ayant conservé les paramètres de clé de compartiment S3 présents dans la configuration du compartiment, vous devez obtenir l'autorisation d'effectuer l'action `s3:GetEncryptionConfiguration`. Pour plus d'informations, consultez GetBucketEncryptionle manuel Amazon Simple Storage Service API Reference.

Utilisation de la console S3

Dans la console S3, vous pouvez afficher les paramètres de clé de compartiment S3 pour votre compartiment ou votre objet. Les paramètres de clé de compartiment S3 sont issus de la configuration du compartiment, sauf si une clé de compartiment S3 est déjà configurée pour les objets source.

Les objets et les dossiers dans le même compartiment peuvent avoir des paramètres de clé de compartiment S3 différents. Par exemple, si vous chargez un objet à l'aide de l'API REST et que vous activez une clé de compartiment S3 pour l'objet, l'objet conserve son paramètre de clé de compartiment S3 dans le compartiment de destination, même si la clé de compartiment S3 est désactivée dans le compartiment de destination. Autre exemple, si vous activez une clé de compartiment S3 pour un compartiment existant, les objets qui se trouvent déjà dans le compartiment n'utilisent pas de clé de compartiment S3. Toutefois, une clé de compartiment S3 est activée pour les nouveaux objets.

Pour afficher le paramètre de clé de compartiment S3 pour votre compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le compartiment pour lequel vous souhaitez activer une clé de compartiment S3.
4. Choisissez Propriétés.
5. Dans la section Chiffrement par défaut, sous Clé de compartiment, vous voyez le paramètre de clé de compartiment S3 pour votre compartiment.

   Si le paramètre de clé de compartiment S3 ne s'affiche pas, il se peut que vous n'ayez pas l'autorisation d'exécuter l'action `s3:GetEncryptionConfiguration`. Pour plus d'informations, consultez GetBucketEncryptionle manuel Amazon Simple Storage Service API Reference.

Pour afficher le paramètre de clé de compartiment S3 pour votre objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le compartiment pour lequel vous souhaitez activer une clé de compartiment S3.
3. Dans la liste Objets, choisissez le nom de votre objet.
4. Sous l'onglet Détails, sous Paramètres de chiffrement côté serveur, choisissez Modifier.

   Sous Clé de compartiment, vous voyez le paramètre de clé de compartiment S3 pour votre objet. Vous ne pouvez pas modifier ce paramètre.

À l'aide du AWS CLI

Pour renvoyer les paramètres de clé de compartiment S3 au niveau du compartiment

Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations.

```
aws s3api get-bucket-encryption --bucket example-s3-bucket1
```

Pour plus d'informations, consultez get-bucket-encryptionle manuel de référence des AWS CLI commandes.

Pour renvoyer les paramètres au niveau de l'objet pour une clé de compartiment S3

Pour utiliser cet exemple, remplacez chaque *user input placeholder* par vos propres informations.

```
aws s3api head-object --bucket example-s3-bucket1 --key my_images.tar.bz2
```

Pour plus d'informations, consultez head-object dans la Référence des commandes AWS CLI .

Utilisation de l'API REST

Pour renvoyer les paramètres de clé de compartiment S3 au niveau du compartiment

Pour renvoyer des informations de chiffrement pour un compartiment, y compris les paramètres d'une clé de compartiment S3, utilisez l'opération `GetBucketEncryption`. Les paramètres de clé de compartiment S3 sont renvoyés dans le corps de la réponse dans l'élément `ServerSideEncryptionConfiguration` avec le paramètre `BucketKeyEnabled`. Pour plus d'informations, consultez GetBucketEncryptionle manuel de référence des API Amazon S3.

Pour renvoyer les paramètres au niveau de l'objet pour une clé de compartiment S3

Pour renvoyer l'état Clé de compartiment S3 d'un objet, utilisez l'opération `HeadObject`. `HeadObject` renvoie l'en-tête de réponse `x-amz-server-side-encryption-bucket-key-enabled` pour indiquer si une clé de compartiment S3 est activée ou désactivée pour l'objet. Pour plus d'informations, consultez HeadObjectle manuel de référence des API Amazon S3.

Les opérations d'API suivantes retournent également l'en-tête de réponse `x-amz-server-side-encryption-bucket-key-enabled` si une clé de compartiment S3 est configurée pour un objet :

- PutObject
- PostObject
- CopyObject
- CreateMultipartUpload
- UploadPartCopy
- UploadPart
- CompleteMultipartUpload
- GetObject

Utilisation du chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS)

L'utilisation du chiffrement double couche côté serveur avec des clés AWS Key Management Service (AWS KMS) (DSSE-KMS) applique deux couches de chiffrement aux objets lorsqu'ils sont chargés sur Amazon S3. DSSE-KMS vous permet de satisfaire plus facilement aux normes de conformité, qui vous imposent d'appliquer un chiffrement multicouche à vos données et de contrôler entièrement vos clés de chiffrement.

Lorsque vous utilisez le DSSE-KMS avec un compartiment Amazon S3, les AWS KMS clés doivent se trouver dans la même région que le compartiment. De même, lorsqu'un chiffrement DSSE-KMS est demandé pour l'objet, le total de contrôle S3, qui fait partie des métadonnées de l'objet, est stocké sous une forme chiffrée. Pour en savoir plus sur le total de contrôle, consultez Vérification de l'intégrité des objets.

L'utilisation de DSSE-KMS et. AWS KMS keys Pour en savoir plus sur la tarification de DSSE-KMS, consultez Concepts de AWS KMS key dans le Guide du développeur AWS Key Management Service et Tarification AWS KMS.

> **Note**
>
> Les clés de compartiment S3 ne sont pas prises en charge pour DSSE-KMS.

Nécessite un chiffrement double couche côté serveur avec AWS KMS keys (DSSE-KMS)

Pour exiger un chiffrement double couche côté serveur de tous les objets contenus dans un compartiment Amazon S3 déterminé, vous pouvez utiliser une stratégie de compartiment. Par exemple, la stratégie de compartiment suivante n'autorise pas le chargement d'objet (`s3:PutObject`) si la demande n'inclut pas d'en-tête `x-amz-server-side-encryption` demandant un chiffrement côté serveur avec DSSE-KMS.

```
{
            "Version":"2012-10-17",
            "Id":"PutObjectPolicy",
            "Statement":[{
                  "Sid":"DenyUnEncryptedObjectUploads",
                  "Effect":"Deny",
                  "Principal":"*",
                  "Action":"s3:PutObject",
```

```
                "Resource":"arn:aws:s3:::example-s3-bucket1/*",
                "Condition":{
                   "StringNotEquals":{
                      "s3:x-amz-server-side-encryption":"aws:kms:dsse"
                   }
                }
             }
          ]
       }
```

Rubriques

- Spécification du chiffrement double couche côté serveur avec des clés AWS KMS (DSSE-KMS)

Spécification du chiffrement double couche côté serveur avec des clés AWS KMS (DSSE-KMS)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets qui sont chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

Si vous souhaitez spécifier un type de chiffrement différent dans vos PUT demandes, vous pouvez utiliser le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-

KMS), le chiffrement double couche côté serveur avec des clés (DSSE-KMS) ou le chiffrement côté serveur avec des AWS KMS clés fournies par le client (SSE-C). Si vous souhaitez définir une autre configuration de chiffrement par défaut dans le compartiment de destination, vous pouvez utiliser SSE-KMS ou DSSE-KMS.

Vous pouvez appliquer le chiffrement lorsque vous chargez un nouvel objet ou copiez un objet existant.

Vous pouvez spécifier DSSE-KMS en utilisant la console Amazon S3, l'API REST Amazon S3 et l' AWS Command Line Interface (AWS CLI). Pour plus d'informations, consultez les rubriques suivantes.

> **Note**
>
> Vous pouvez utiliser plusieurs régions AWS KMS keys dans Amazon S3. Cependant, Amazon S3 traite actuellement les clés multi-régions comme s'il s'agissait de clés à région unique et n'utilise pas les fonctions multi-régions de la clé. Pour en savoir plus, consultez la section Utilisation des clés multi-régions dans le Guide du développeur AWS Key Management Service .

> **Note**
>
> Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez avoir l'autorisation d'utiliser la clé. Pour plus d'informations sur les autorisations intercomptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service .

Utilisation de la console S3

Cette section explique comment définir ou modifier le type de chiffrement d'un objet afin d'utiliser un chiffrement double couche côté serveur avec des clés AWS Key Management Service (AWS KMS) (DSSE-KMS) à l'aide de la console Amazon S3.

> **Note**
>
> - Si vous changez la méthode de chiffrement d'un objet, un objet est créé en remplacement de l'ancien. Si la gestion des versions S3 est activée, une nouvelle version de l'objet est créée et l'objet existant devient une version plus ancienne. Le rôle qui modifie la propriété devient également le propriétaire du nouvel objet ou (version de l'objet).
> - Si vous modifiez le type de chiffrement d'un objet doté de balises définies par l'utilisateur, vous devez disposer de cette `s3:GetObjectTagging` autorisation. Si vous modifiez le type de chiffrement d'un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`s3:GetObjectTagging`autorisation.
>
>   Si la politique du compartiment de destination refuse l'`s3:GetObjectTagging`action, le type de chiffrement de l'objet sera mis à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

Pour ajouter ou modifier le chiffrement d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment qui contient l'objet à chiffrer.
4. Dans la liste Objets, cochez la case correspondant à l'objet pour lequel vous souhaitez ajouter ou modifier le chiffrement.

   La page de détails de l'objet apparaît, avec plusieurs sections qui affichent les propriétés de votre objet.
5. Choisissez l’onglet Propriétés.
6. Faites défiler la page vers le bas jusqu'à la section Chiffrement par défaut, puis choisissez Modifier.

   La page Modifier le chiffrement par défaut s'ouvre.
7. Sous Type de chiffrement, choisissez Chiffrement double couche côté serveur avec AWS Key Management Service clés (DSSE-KMS).
8. Sous CléAWS KMS , choisissez votre clé KMS avec l'une des options suivantes :

- Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis sélectionnez votre Clé KMS dans la liste des clés disponibles.

  La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .
- Pour saisir l'ARN de la clé KMS, choisissez Enter AWS KMS key ARN, puis entrez l'ARN de votre clé KMS dans le champ qui apparaît.
- Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

  Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous pouvez uniquement utiliser des clés KMS disponibles dans la même Région AWS que le compartiment. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé, puis vous devez saisir l'ARN de la clé KMS.
> Amazon S3 prend en charge seulement les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour en savoir plus, consultez Identification des clés KMS asymétriques dans le Guide du développeur AWS Key Management Service .

9. Pour Clé de compartiment, choisissez Désactiver. Les clés de compartiment S3 ne sont pas prises en charge pour DSSE-KMS.
10. Sélectionnez Save Changes (Enregistrer les modifications).

> **Note**
>
> Cette action applique le chiffrement à tous les objets spécifiés. Lorsque vous chiffrez des dossiers, attendez la fin de l'opération d'enregistrement pour ajouter de nouveaux objets au dossier.

Utilisation de l'API REST

Lorsque vous créez un objet, c'est-à-dire lorsque vous téléchargez un nouvel objet ou que vous copiez un objet existant, vous pouvez spécifier l'utilisation du chiffrement double couche côté serveur AWS KMS keys (DSSE-KMS) pour chiffrer vos données. Pour ce faire, ajoutez l'en-tête `x-amz-server-side-encryption` à la demande. Configurez la valeur de l'en-tête sur l'algorithme de chiffrement `aws:kms:dsse`. Amazon S3 confirme que votre objet est stocké avec un chiffrement DSSE-KMS en renvoyant l'en-tête de réponse `x-amz-server-side-encryption`.

Si vous spécifiez l'en-tête `x-amz-server-side-encryption` avec une valeur de `aws:kms:dsse`, vous pouvez également utiliser les en-têtes de demandes suivants :

- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`
- `x-amz-server-side-encryption-context: SSEKMSEncryptionContext`

Rubriques

- Opérations de l'API REST Amazon S3 prenant en charge DSSE-KMS
- Contexte de chiffrement (x-amz-server-side-encryption-context)
- AWS KMS ID de clé (x-amz-server-side-encryption-aws-kms-key-id)

Opérations de l'API REST Amazon S3 prenant en charge DSSE-KMS

Les opérations d'API REST suivantes acceptent les en-têtes de demande `x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-id` et `x-amz-server-side-encryption-context`.

- PutObject – Lorsque vous chargez des données avec l'opération d'API PUT, vous pouvez spécifier ces en-têtes de demande.
- CopyObject – Lorsque vous copiez un objet, vous disposez d'un objet source et d'un objet cible. Lorsque vous transmettez des en-têtes DSSE-KMS avec l'opération `CopyObject`, ils s'appliquent

uniquement à l'objet cible. Lorsque vous copiez un objet existant, que l'objet source soit chiffré ou non, l'objet de destination n'est pas chiffré sauf si vous demandez explicitement un chiffrement côté serveur.

- Objet POST – Lorsque vous utilisez une opération `POST` pour charger un objet, plutôt que des en-têtes de demande, vous fournissez les mêmes informations dans les champs du formulaire.
- CreateMultipartUpload – Lorsque vous chargez des objets volumineux en procédant à un chargement partitionné, vous pouvez spécifier ces en-têtes dans la demande `CreateMultipartUpload`.

Les en-têtes de réponse des opérations d'API REST suivantes renvoient l'en-tête `x-amz-server-side-encryption` lorsqu'un objet est stocké avec un chiffrement côté serveur.

- PutObject
- CopyObject
- Objet POST
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Toutes `GET` les `PUT` demandes relatives à un objet protégé par AWS KMS échouent si vous ne les créez pas à l'aide du protocole SSL (Secure Sockets Layer), du protocole TLS (Transport Layer Security) ou de la version 4 de signature.
> - Si votre objet utilise DSSE-KMS, n'envoyez pas d'en-têtes de demande de chiffrement pour les demandes `GET` et les demandes `HEAD`, car vous obtiendrez une erreur HTTP 400 (Demande incorrecte).

Contexte de chiffrement (**`x-amz-server-side-encryption-context`**)

Si vous spécifiez `x-amz-server-side-encryption:aws:kms:dsse`, l'API Simple Storage Service (Amazon S3) prend en charge un contexte de chiffrement avec l'en-tête `x-amz-server-side-encryption-context`. Un contexte de chiffrement est un ensemble de paires valeur clé qui contient des informations contextuelles supplémentaires sur les données.

Amazon S3 utilise automatiquement l'Amazon Resource Name (ARN) de l'objet en tant que paire de contexte de chiffrement ; par exemple, `arn:aws:s3:::`*`object_ARN`*.

Vous pouvez éventuellement fournir une paire de contexte de chiffrement supplémentaire à l'aide de l'en-tête `x-amz-server-side-encryption-context`. Toutefois, étant donné que le contexte de chiffrement n'est pas chiffré, assurez-vous qu'il n'inclut pas d'informations sensibles. Amazon S3 stocke cette paire de clés supplémentaire avec le contexte de chiffrement par défaut.

Pour plus d'informations sur le contexte de chiffrement dans Simple Storage Service (Amazon S3), consultez Contexte de chiffrement. Pour des informations générales sur le contexte de chiffrement, consultez la section Concepts AWS Key Management Service - Contexte de chiffrement du Guide du développeur AWS Key Management Service .

AWS KMS ID de clé (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Vous pouvez utiliser l'en-tête `x-amz-server-side-encryption-aws-kms-key-id` pour spécifier l'ID de la clé gérée par le client utilisée pour protéger les données. Si vous spécifiez l'`x-amz-server-side-encryption:aws:kms:dsse`en-tête mais ne le `x-amz-server-side-encryption-aws-kms-key-id` fournissez pas, Amazon S3 utilise le Clé gérée par AWS (`aws/s3`) pour protéger les données. Si vous souhaitez utiliser une clé gérée par le client, vous devrez fournir l'en-tête `x-amz-server-side-encryption-aws-kms-key-id` de la clé gérée par le client.

> **Important**
>
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 ne prend en charge que les clés KMS à chiffrement symétrique. Pour plus d'informations sur ces clés, consultez Clés KMS de chiffrement symétriques dans le Guide du développeur AWS Key Management Service .

À l'aide du AWS CLI

Lorsque vous chargez un nouvel objet ou copiez un objet existant, vous pouvez spécifier l'utilisation de DSSE-KMS pour chiffrer vos données. Pour ce faire, ajoutez le paramètre `--server-side-encryption aws:kms:dsse` à l'en-tête. Utilisez le paramètre `--ssekms-key-id` *`example-key-id`* pour ajouter la clé AWS KMS gérée par le client que vous avez créée. Si vous spécifiez`--server-side-encryption aws:kms:dsse`, mais que vous ne fournissez pas d'identifiant de AWS KMS clé, Amazon S3 utilisera la clé AWS gérée (`aws/s3`).

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --server-
side-encryption aws:kms:dsse --ssekms-key-id example-key-id --body filepath
```

Vous pouvez chiffrer un objet non chiffré pour utiliser DSSE-KMS en recopiant l'objet à son emplacement.

```
aws s3api copy-object --bucket DOC-EXAMPLE-BUCKET --key example-object-key --
body filepath --bucket DOC-EXAMPLE-BUCKET --key example-object-key --sse aws:kms:dsse
 --sse-kms-key-id example-key-id --body filepath
```

Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C)

Le chiffrement côté serveur consiste à protéger les données au repos. Un chiffrement côté serveur chiffre uniquement les données d'objet, pas les métadonnées d'objet. En utilisant le chiffrement côté serveur avec des clés fournies par le client (SSE-C), vous pouvez stocker vos données chiffrées avec vos propres clés de chiffrement. Avec la clé de chiffrement que vous fournissez dans la demande, Amazon S3 gère le chiffrement des données quand il écrit sur les disques et le déchiffrement des données quand vous accédez à vos objets. Par conséquent, vous n'avez pas besoin de conserver de code pour procéder au chiffrement et déchiffrement des données. Il ne vous reste qu'à gérer les clés de chiffrement que vous fournissez.

Quand vous chargez un objet, Amazon S3 utilise la clé de chiffrement que vous fournissez pour appliquer le chiffrement AES-256 à vos données. Amazon S3 supprime ensuite la clé de chiffrement de la mémoire. Lorsque vous récupérez un objet, vous devez fournir la même clé de chiffrement dans la demande. Amazon S3 vérifie tout d'abord que la clé de chiffrement que vous avez fournie correspond, puis il déchiffre l'objet avant de vous renvoyer les données de ce dernier.

Aucuns frais supplémentaires ne s'appliquent à l'utilisation du chiffrement SSE-C. Toutefois, les demandes de configuration et d'utilisation du chiffrement SSE-C entraînent des frais de demande

Amazon S3 standard. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3.

> **Note**
>
> Amazon S3 ne stocke pas la clé de chiffrement que vous fournissez. À la place, il stocke un code d'authentification de message utilisant hash (HMAC) crypté de manière aléatoire de la clé de chiffrement pour valider les demandes futures. La valeur HMAC cryptée ne peut pas être utilisée pour retrouver la valeur de la clé de chiffrement ou pour déchiffrer les contenus de l'objet chiffré. Cela signifie que si vous perdez la clé de chiffrement, vous perdez l'objet.

S3 Replication prend en charge les objets chiffrés avec le SSE-C. Pour plus d'informations sur la réplication des objets chiffrés, consultez the section called “Réplication d'objets chiffrés”.

Pour plus d'informations sur SSE-C, consultez les rubriques suivantes :

Rubriques

- Présentation des SSE-C
- Exigence et restriction des SSE-C
- URL présignées et SSE-C
- Spécification du chiffrement côté serveur avec des clés fournies par le client (SSE-C)

Présentation des SSE-C

Cette section fournit une présentation du chiffrement SSE-C. Dans le cadre du chiffrement SSE-C, gardez à l'esprit les considérations suivantes.

- Vous devez utiliser HTTPS.

  > **Important**
  >
  > Amazon S3 rejette toute demande faite via HTTP lors de l'utilisation du chiffrement SSE-C. Pour des raisons de sécurité, nous vous recommandons de considérer toute clé que vous envoyez par erreur via HTTP comme compromise. Écartez la clé et permutez comme il convient.

- La balise d'entité (ETag) dans la réponse n'est pas le hachage MD5 des données de l'objet.

- Vous gérez un mappage pour savoir quelle clé de chiffrement a été utilisée pour chiffrer quel objet. Amazon S3 ne stocke pas les clés de chiffrement. Vous devez assurer le suivi pour savoir quelle clé de chiffrement a été fournie pour quel objet.
  - Si votre compartiment prend en charge la gestion des versions, chaque version d'objet que vous chargez à l'aide de cette fonctionnalité peut avoir sa propre clé de chiffrement. Vous devez assurer le suivi pour savoir quelle clé de chiffrement a été utilisée pour quelle version d'objet.
  - Etant donné que vous gérez les clés de chiffrement du côté client, vous gérez toute sauvegarde supplémentaire, comme la rotation des clés, du côté client.

> **Warning**
>
> Si vous perdez la clé de chiffrement, toute requête `GET` d'un objet sans clé de chiffrement échoue et vous perdez l'objet.

Exigence et restriction des SSE-C

Pour exiger le chiffrement SSE-C de tous les objets figurant dans un compartiment Amazon S3 particulier, vous pouvez utiliser une politique de compartiment.

Par exemple, la stratégie de compartiment suivante refuse les autorisations de chargement d'objet (`s3:PutObject`) autorisations pour toutes les demandes qui n'incluent pas l'en-tête `x-amz-server-side-encryption-customer-algorithm` demandant SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RequireSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
```

```
    ]
}
```

Vous pouvez également utiliser une politique pour limiter le chiffrement côté serveur de tous les objets figurant dans un compartiment Amazon S3 particulier. Par exemple, la politique de compartiment suivante n'autorise pas le chargement d'objet (`s3:PutObject`) si la demande inclut l'en-tête `x-amz-server-side-encryption-customer-algorithm` demandant le chiffrement SSE-C.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
                }
            }
        }
    ]
}
```

> **Important**
>
> Si vous utilisez une politique de compartiment pour activer SSE-C`s3:PutObject`, vous devez inclure l'`x-amz-server-side-encryption-customer-algorithm`en-tête dans toutes les demandes de téléchargement partitionné (CreateMultipartUpload UploadPart, et). CompleteMultipartUpload

URL présignées et SSE-C

Vous pouvez générer une URL pré-signée qui peut être utilisée pour des opérations comme le chargement d'un nouvel objet, la récupération d'un objet existant ou la récupération des métadonnées d'objet. Les URL pré-signées prennent en charge le chiffrement SSE-C comme suit :

- Lors de la création d'une URL pré-signée, vous devez spécifier l'algorithme en utilisant l'en-tête `x-amz-server-side-encryption-customer-algorithm` dans le calcul de la signature.
- Lorsque vous utilisez l'URL pré-signée pour charger un nouvel objet, récupérer un objet existant ou récupérer uniquement des métadonnées d'objet, vous devez fournir tous les en-têtes de chiffrement dans la demande de votre application cliente.

> **Note**
>
> Pour des objets hors chiffrement SSE-C, vous pouvez générer une URL pré-signée et la coller directement dans un navigateur pour accéder aux données.
> Toutefois, vous ne pouvez pas procéder ainsi pour des objets SSE-C, car en plus de l'URL pré-signée, vous devez également inclure des en-têtes HTTP spécifiques aux objets SSE-C. Par conséquent, vous pouvez utiliser des URL pré-signées pour les objets SSE-C uniquement par programmation.

Pour en savoir plus sur les URL présignées, veuillez consulter the section called “Utilisation d'URL présignées”.

Spécification du chiffrement côté serveur avec des clés fournies par le client (SSE-C)

Au moment de la création de l'objet avec l'API REST, vous pouvez spécifier un chiffrement côté serveur avec les clés fournies par le client (SSE-C). Lorsque vous utilisez SSE-C, vous devez fournir des informations de clé de chiffrement à l'aide des en-têtes de demande suivants.

Nom	Description
`x-amz-server-side-encryption-customer-algorithm`	Utilisez cet en-tête pour spécifier l'algorithme du chiffrement. La valeur de l'en-tête doit être `AES256`.

Nom	Description
`x-amz-server-side-encryption-customer-key`	Utilisez cet en-tête pour fournir la clé de chiffrement de 256 bits encodée en Base64 qu'Amazon S3 doit utiliser pour chiffrer ou déchiffrer les données.
`x-amz-server-side-encryption-customer-key-MD5`	Utilisez cet en-tête pour fournir la valeur de hachage MD5 128 bits encodée en Base64 de la clé de chiffrement conformément à la norme RFC 1321. Amazon S3 utilise cet en-tête pour vérifier l'intégrité du message et veiller à ce que la clé de chiffrement ait été transmise sans erreur.

Vous pouvez utiliser les bibliothèques d'encapsulation du AWS SDK pour ajouter ces en-têtes à votre demande. Si nécessaire, vous pouvez effectuer les appels à l'API REST Amazon S3 directement dans l'application.

> **Note**
>
> Vous ne pouvez pas utiliser la console Amazon S3 pour charger un objet et demander des clés SSE-C. Vous ne pouvez pas non plus utiliser la console pour mettre à jour (par exemple, modifier la classe de stockage ou ajouter des métadonnées) un objet existant stocké à l'aide de clés SSE-C.

Utilisation de l'API REST

API REST Amazon S3 prenant en charge SSE-C

Les API Amazon S3 suivantes prennent en charge le chiffrement côté serveur avec les clés de chiffrement fournies par le client (SSE-C).

- Opération GET – Lorsque vous récupérez des objets via l'API GET (consultez GetObject), vous pouvez spécifier ces en-têtes de demande.
- Opération HEAD – Pour récupérer les métadonnées d'objet via l'API HEAD (consultez HeadObject), vous pouvez spécifier ces en-têtes de demande.
- Opération PUT – Lorsque vous chargez des données via l'API PutObject (consultez PutObject), vous pouvez spécifier ces en-têtes de demande.

- Chargement partitionné – Lorsque vous chargez des objets volumineux via l'API de chargement partitionné, vous pouvez spécifier ces en-têtes. Vous spécifiez ces en-têtes dans la demande initiale (consultez Lancement du chargement partitionné (langue française non garantie)) et dans chaque demande de chargement de partie suivante (consultez Chargement d'une partie ou Chargement d'une partie (Copy) (langue française non garantie)). Pour chaque demande de chargement d'une partie, les informations de chiffrement doivent être les mêmes que celles fournies dans la demande de lancement du chargement partitionné.
- Opération POST – Lorsque vous utilisez une opération POST pour charger un objet (consultez Objet POST), à la place des en-têtes de demande, fournissez les mêmes informations dans les champs du formulaire.
- Opération Copy – Lorsque vous copiez un objet (consultez CopyObject), vous disposez d'un objet source et d'un objet cible :
  - Si vous souhaitez que l'objet cible soit chiffré à l'aide d'un chiffrement côté serveur avec des clés AWS gérées, vous devez fournir l'en-tête de `x-amz-server-side-encryption` demande.
  - Si vous souhaitez chiffrer l'objet cible grâce aux SSE-C, vous devez fournir des informations de chiffrement grâce aux trois en-têtes décrits dans le tableau précédent.
  - Si l'objet source est chiffré grâce aux SSE-C, vous devez fournir les informations sur la clé de chiffrement grâce aux en-têtes suivants afin qu'Amazon S3 puisse déchiffrer l'objet pour le copier.

Nom	Description
`x-amz-copy-source-server-side-encryption-customer-algorithm`	Incluez cet en-tête pour spécifier l'algorithme qu'Amazon S3 doit utiliser pour déchiffrer l'objet source. La valeur doit être `AES256`.
`x-amz-copy-source-server-side-encryption-customer-key`	Incluez cet en-tête pour fournir la clé de chiffrement encodée en Base64 qu'Amazon S3 doit utiliser pour déchiffrer l'objet source. La clé de chiffrement doit être celle fournie à Amazon S3 lorsque vous avez créé l'objet source. Sinon, Amazon S3 ne peut pas déchiffrer l'objet.

Nom	Description
`x-amz-copy-source-server-side-encryption-customer-key-MD5`	Incluez cet en-tête pour fournir la valeur de hachage MD5 128 bits encodée en Base64 de la clé de chiffrement conformément à la norme RFC 1321.

Utilisation des AWS SDK pour spécifier le SSE-C pour les opérations PUT, GET, Head et Copy

Les exemples suivants illustrent la demande d'un chiffrement côté serveur avec les clés de chiffrement fournies par le client (SSE-C) pour les objets. Les exemples exécutent les opérations suivantes. Chaque opération montre comment spécifier les en-têtes SSE-C dans la demande :

- Put object – Charge un objet et demande un chiffrement côté serveur avec une clé de chiffrement fournie par le client.
- Get object – Télécharge l'objet chargé à l'étape précédente. Dans la demande, vous fournissez les mêmes informations de chiffrement que celles fournies lors du chargement de l'objet. Amazon S3 a besoin de ces informations pour déchiffrer l'objet afin de pouvoir vous le renvoyer.
- Get object metadata – Récupère les métadonnées de l'objet. Vous fournissez les mêmes informations de chiffrement que celles utilisées quand l'objet a été chargé.
- Copy object – Effectue une copie de l'objet précédemment chargé. Comme l'objet source est stocké via SSE-C, vous devez fournir ses informations de chiffrement dans votre demande de copie. Par défaut, Amazon S3 ne chiffre la copie de l'objet que si vous le demandez explicitement. Cet exemple demande à Amazon S3 de stocker une copie chiffrée de l'objet.

Java

> **Note**
>
> Cet exemple montre comment copier un objet en une seule opération. Lorsque vous utilisez l'API de chargement partitionné pour charger des objets volumineux, vous fournissez les informations de chiffrement comme illustré dans l'exemple suivant. Pour des exemples de téléchargements partitionnés utilisant le AWS SDK for Java, voir. Chargement d'un objet à l'aide du chargement partitionné

Pour ajouter les informations de chiffrement requises, vous incluez une clé `SSECustomerKey` dans votre demande. Pour plus d'informations sur la classe `SSECustomerKey`, consultez la section API REST.

Pour plus d'informations sur SSE-C, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C). Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import javax.crypto.KeyGenerator;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;

    public static void main(String[] args) throws IOException,
 NoSuchAlgorithmException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String uploadFileName = "*** File path ***";
        String targetKeyName = "*** Target key name ***";

        // Create an encryption key.
        KEY_GENERATOR = KeyGenerator.getInstance("AES");
        KEY_GENERATOR.init(256, new SecureRandom());
```

```
        SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

        try {
            S3_CLIENT = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Upload an object.
            uploadObject(bucketName, keyName, new File(uploadFileName));

            // Download the object.
            downloadObject(bucketName, keyName);

            // Verify that the object is properly encrypted by attempting to
retrieve it
            // using the encryption key.
            retrieveObjectMetadata(bucketName, keyName);

            // Copy the object into a new object that also uses SSE-C.
            copyObject(bucketName, keyName, targetKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObject(String bucketName, String keyName, File file) {
        PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
        S3_CLIENT.putObject(putRequest);
        System.out.println("Object uploaded");
    }

    private static void downloadObject(String bucketName, String keyName) throws
IOException {
        GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
        S3Object object = S3_CLIENT.getObject(getObjectRequest);
```

```
        System.out.println("Object content: ");
        displayTextInputStream(object.getObjectContent());
    }

    private static void retrieveObjectMetadata(String bucketName, String keyName) {
        GetObjectMetadataRequest getMetadataRequest = new
 GetObjectMetadataRequest(bucketName, keyName)
                .withSSECustomerKey(SSE_KEY);
        ObjectMetadata objectMetadata =
 S3_CLIENT.getObjectMetadata(getMetadataRequest);
        System.out.println("Metadata retrieved. Object size: " +
 objectMetadata.getContentLength());
    }

    private static void copyObject(String bucketName, String keyName, String
 targetKeyName)
            throws NoSuchAlgorithmException {
        // Create a new encryption key for target so that the target is saved using
        // SSE-C.
        SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

        CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
 bucketName, targetKeyName)
                .withSourceSSECustomerKey(SSE_KEY)
                .withDestinationSSECustomerKey(newSSEKey);

        S3_CLIENT.copyObject(copyRequest);
        System.out.println("Object copied");
    }

    private static void displayTextInputStream(S3ObjectInputStream input) throws
 IOException {
        // Read one line at a time from the input stream and display each line.
        BufferedReader reader = new BufferedReader(new InputStreamReader(input));
        String line;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
        System.out.println();
    }
}
```

.NET

> **Note**
>
> Pour obtenir des exemples de chargement d'objets volumineux à l'aide de l'API de chargement partitionné, consultez Chargement d'un objet à l'aide du chargement partitionné et Utilisation des AWS SDK (API de bas niveau).

Pour plus d'informations sur SSE-C, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C). Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for new object created ***";
        private const string copyTargetKeyName = "*** key name for object copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            ObjectOpsUsingClientEncryptionKeyAsync().Wait();
        }
        private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
```

```
        {
            try
            {
                // Create an encryption key.
                Aes aesEncryption = Aes.Create();
                aesEncryption.KeySize = 256;
                aesEncryption.GenerateKey();
                string base64Key = Convert.ToBase64String(aesEncryption.Key);

                // 1. Upload the object.
                PutObjectRequest putObjectRequest = await
UploadObjectAsync(base64Key);
                // 2. Download the object and verify that its contents matches what
you uploaded.
                await DownloadObjectAsync(base64Key, putObjectRequest);
                // 3. Get object metadata and verify that the object uses AES-256
encryption.
                await GetObjectMetadataAsync(base64Key);
                // 4. Copy both the source and target objects using server-side
encryption with
                //    a customer-provided encryption key.
                await CopyObjectAsync(aesEncryption, base64Key);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        private static async Task<PutObjectRequest> UploadObjectAsync(string
base64Key)
        {
            PutObjectRequest putObjectRequest = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                ContentBody = "sample text",
```

```
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };
            PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
            return putObjectRequest;
        }
        private static async Task DownloadObjectAsync(string base64Key,
PutObjectRequest putObjectRequest)
        {
            GetObjectRequest getObjectRequest = new GetObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                // Provide encryption information for the object stored in Amazon
S3.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
            using (StreamReader reader = new
StreamReader(getResponse.ResponseStream))
            {
                string content = reader.ReadToEnd();
                if (String.Compare(putObjectRequest.ContentBody, content) == 0)
                    Console.WriteLine("Object content is same as we uploaded");
                else
                    Console.WriteLine("Error...Object content is not same.");

                if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
                    Console.WriteLine("Object encryption method is AES256, same as
we set");
                else
                    Console.WriteLine("Error...Object encryption method is not the
same as AES256 we set");

                // Assert.AreEqual(putObjectRequest.ContentBody, content);
                // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getResponse.ServerSideEncryptionCustomerMethod);
```

```
            }
        }
        private static async Task GetObjectMetadataAsync(string base64Key)
        {
            GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
            {
                BucketName = bucketName,
                Key = keyName,

                // The object stored in Amazon S3 is encrypted, so provide the
necessary encryption information.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
            Console.WriteLine("The object metadata show encryption method used is:
{0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
            // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
        }
        private static async Task CopyObjectAsync(Aes aesEncryption, string
base64Key)
        {
            aesEncryption.GenerateKey();
            string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

            CopyObjectRequest copyRequest = new CopyObjectRequest
            {
                SourceBucket = bucketName,
                SourceKey = keyName,
                DestinationBucket = bucketName,
                DestinationKey = copyTargetKeyName,
                // Information about the source object's encryption.
                CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
                // Information about the target object's encryption.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = copyBase64Key
```

```
            };
            await client.CopyObjectAsync(copyRequest);
        }
    }
}
```

Utilisation des AWS SDK pour spécifier le SSE-C pour les téléchargements partitionnés

L'exemple de la section précédente montre comment demander le chiffrement côté serveur à l'aide d'une clé de chiffrement fournie par le client (SSE-C) dans les opérations PUT, GET, Head et Copy. Cette section décrit d'autres API Amazon S3 prenant en charge SSE-C.

Java

Pour charger des objets volumineux, vous pouvez utiliser l'API de chargement partitionné (consultez Chargement et copie d'objets à l'aide d'un chargement partitionné). Auquel cas, vous avez le choix entre l'API de haut niveau ou l'API de bas niveau. Ces API permettent l'utilisation d'en-têtes liés au chiffrement dans votre demande.

- Lorsque vous utilisez l'API `TransferManager` de haut niveau , vous fournissez les en-têtes propres au chiffrement dans l'objet `PutObjectRequest` (voir Chargement d'un objet à l'aide du chargement partitionné).
- Lorsque vous utilisez l'API de bas niveau, vous fournissez les informations de chiffrement dans l'objet `InitiateMultipartUploadRequest`, suivies des mêmes informations dans chaque `UploadPartRequest`. Il est inutile de fournir des en-têtes liés au chiffrement dans votre objet `CompleteMultipartUploadRequest`. Pour obtenir des exemples, consultez Utilisation des AWS SDK (API de bas niveau).

L'exemple ci-dessous utilise `TransferManager` pour créer des objets ; il explique également la marche à suivre pour fournir des informations SSE-C. Cet exemple effectue les opérations suivantes :

- Crée un objet à l'aide de la méthode `TransferManager.upload()`. Dans l'instance `PutObjectRequest`, vous fournissez à la demande les informations de clé de chiffrement. Amazon S3 chiffre l'objet en utilisant la clé fournie par le client.
- Effectue une copie de l'objet en appelant la méthode `TransferManager.copy()`. L'exemple demande à Amazon S3 de chiffrer la copie de l'objet à l'aide d'un nouvel objet `SSECustomerKey`. L'objet source étant chiffré au moyen de SSE-C,

`CopyObjectRequest`fournit également la clé de chiffrement de l'objet source, afin qu'Amazon S3 puisse déchiffrer l'objet avant de le copier.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.SSECustomerKey;
import com.amazonaws.services.s3.transfer.Copy;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import javax.crypto.KeyGenerator;
import java.io.File;
import java.security.SecureRandom;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String fileToUpload = "*** File path ***";
        String keyName = "*** New object key name ***";
        String targetKeyName = "*** Key name for object copy ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();
            TransferManager tm = TransferManagerBuilder.standard()
                    .withS3Client(s3Client)
                    .build();

            // Create an object from a file.
```

```
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName,
keyName, new File(fileToUpload));

            // Create an encryption key.
            KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
            keyGenerator.init(256, new SecureRandom());
            SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

            // Upload the object. TransferManager uploads asynchronously, so this
call
            // returns immediately.
            putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
            Upload upload = tm.upload(putObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            upload.waitForCompletion();
            System.out.println("Object created.");

            // Copy the object and store the copy using SSE-C with a new key.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
            SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
            copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);

copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

            // Copy the object. TransferManager copies asynchronously, so this call
returns
            // immediately.
            Copy copy = tm.copy(copyObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            copy.waitForCompletion();
            System.out.println("Copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
```

```
        }
    }
}
```

.NET

Pour télécharger des objets volumineux, vous pouvez utiliser l'API de téléchargement partitionné (voirChargement et copie d'objets à l'aide d'un chargement partitionné). AWS Le SDK for .NET fournit des API de haut niveau ou de bas niveau pour télécharger des objets volumineux. Ces API permettent l'utilisation d'en-têtes liés au chiffrement dans votre demande.

- Lorsque vous utilisez l'API de haut niveau `Transfer-Utility` , vous fournissez les en-têtes propres au chiffrement dans l'objet `TransferUtilityUploadRequest` comme illustré. Pour des exemples de code, consultez Chargement d'un objet à l'aide du chargement partitionné.

  ```
  TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
  {
      FilePath = filePath,
      BucketName = existingBucketName,
      Key = keyName,
      // Provide encryption information.
      ServerSideEncryptionCustomerMethod =
   ServerSideEncryptionCustomerMethod.AES256,
      ServerSideEncryptionCustomerProvidedKey = base64Key,
  };
  ```

- Lorsque vous utilisez l'API de bas niveau, vous fournissez des informations de chiffrement dans votre demande de lancement du chargement partitionné. Ces informations de chiffrement doivent être identiques dans les demandes de chargement partitionné qui suivent. Il est toutefois inutile de fournir des en-têtes liés au chiffrement dans votre demande de fin de chargement partitionné. Pour obtenir des exemples, consultez Utilisation des AWS SDK (API de bas niveau).

  L'exemple suivant concerne un chargement partitionné de niveau inférieur exécutant une copie d'un objet volumineux existant. Dans cet exemple, l'objet à copier est stocké dans Amazon S3 à l'aide d'une clé SSE-C, et vous souhaitez également enregistrer l'objet cible à l'aide d'une clé SSE-C. Dans l'exemple, vous procédez comme suit :

  - Initier une demande de chargement partitionné en fournissant une clé de chiffrement et les informations connexes.

- Fournir les clés de chiffrement de l'objet source et cible et les informations relatives dans la `CopyPartRequest`.
- Obtenir la taille de l'objet source à copier en récupérant les métadonnées de l'objet.
- Charger les objets par lots de 5 Mo.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUcopyObjectTest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string sourceKeyName      = "*** source object key name
 ***";
        private const string targetKeyName      = "*** key name for the target
 object ***";
        private const string filePath           = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CopyObjClientEncryptionKeyAsync().Wait();
        }

        private static async Task CopyObjClientEncryptionKeyAsync()
        {
            Aes aesEncryption = Aes.Create();
            aesEncryption.KeySize = 256;
            aesEncryption.GenerateKey();
            string base64Key = Convert.ToBase64String(aesEncryption.Key);
```

```
            await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key,
s3Client);

            await CopyObjectAsync(s3Client, base64Key);
        }
        private static async Task CopyObjectAsync(IAmazonS3 s3Client, string
base64Key)
        {
            List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();

            // 1. Initialize.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
            long firstByte = 0;
            long lastByte = partSize;

            try
            {
                // First find source object size. Because object is stored
encrypted with
                // customer provided key you need to provide encryption
information in your request.
                GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest()
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
```

```
                    ServerSideEncryptionCustomerProvidedKey = base64Key // " *
**source object encryption key ***"
                };

                GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

                long filePosition = 0;
                for (int i = 1; filePosition <
getObjectMetadataResponse.ContentLength; i++)
                {
                    CopyPartRequest copyPartRequest = new CopyPartRequest
                    {
                        UploadId = initResponse.UploadId,
                        // Source.
                        SourceBucket = existingBucketName,
                        SourceKey = sourceKeyName,
                        // Source object is stored using SSE-C. Provide encryption
information.
                        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //"***source object encryption key ***",
                        FirstByte = firstByte,
                        // If the last part is smaller then our normal part size
then use the remaining size.
                        LastByte = lastByte >
getObjectMetadataResponse.ContentLength ?
                            getObjectMetadataResponse.ContentLength - 1 :
lastByte,

                        // Target.
                        DestinationBucket = existingBucketName,
                        DestinationKey = targetKeyName,
                        PartNumber = i,
                        // Encryption information for the target object.
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };
                    uploadResponses.Add(await
s3Client.CopyPartAsync(copyPartRequest));
                    filePosition += partSize;
                    firstByte += partSize;
```

```
                lastByte += partSize;
            }

            // Step 3: complete.
            CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                UploadId = initResponse.UploadId,
            };
            completeRequest.AddPartETags(uploadResponses);

            CompleteMultipartUploadResponse completeUploadResponse =
                await s3Client.CompleteMultipartUploadAsync(completeRequest);
        }
        catch (Exception exception)
        {
            Console.WriteLine("Exception occurred: {0}", exception.Message);
            AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                UploadId = initResponse.UploadId
            };
            s3Client.AbortMultipartUpload(abortMPURequest);
        }
    }
    private static async Task
CreateSampleObjUsingClientEncryptionKeyAsync(string base64Key, IAmazonS3
s3Client)
    {
        // List to store upload part responses.
        List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

        // 1. Initialize.
        InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
        {
            BucketName = existingBucketName,
            Key = sourceKeyName,
```

```
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            InitiateMultipartUploadResponse initResponse =
               await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                    {
                        BucketName = existingBucketName,
                        Key = sourceKeyName,
                        UploadId = initResponse.UploadId,
                        PartNumber = i,
                        PartSize = partSize,
                        FilePosition = filePosition,
                        FilePath = filePath,
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };

                    // Upload part and add response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }

                // Step 3: complete.
                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
```

```
                    UploadId = initResponse.UploadId,
                    //PartETags = new List<PartETag>(uploadResponses)

                };
                completeRequest.AddPartETags(uploadResponses);

                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);

            }
            catch (Exception exception)
            {
                Console.WriteLine("Exception occurred: {0}", exception.Message);
                AbortMultipartUploadRequest abortMPURequest = new
 AbortMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    UploadId = initResponse.UploadId
                };
                await s3Client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
    }
}
```

Protection des données avec le chiffrement côté client

Le chiffrement côté client est le fait de chiffrer vos données localement pour garantir leur sécurité en transit et au repos. Pour chiffrer vos objets avant de les envoyer à Amazon S3, utilisez le client de chiffrement Amazon S3. Lorsque vos objets sont chiffrés de cette manière, ils ne sont exposés à aucun tiers, y compris AWS. Amazon S3 reçoit vos objets déjà chiffrés. Amazon S3 ne joue aucun rôle dans le chiffrement ou le déchiffrement de vos objets. Vous pouvez utiliser à la fois le client de chiffrement Amazon 3 et le chiffrement côté serveur pour chiffrer vos données. Lorsque vous envoyez des objets chiffrés à Amazon S3, Amazon S3 ne reconnaît pas les objets comme étant chiffrés, il ne détecte que des objets classiques.

Le client de chiffrement Amazon S3 fait office d'intermédiaire entre vous et Amazon S3. Une fois que vous avez instancié le client de chiffrement Amazon S3, vos objets sont automatiquement chiffrés et déchiffrés dans le cadre de vos demandes Amazon S3 `PutObject` et `GetObject`. Vos

objets sont tous chiffrés à l'aide d'une clé de données unique. Le client de chiffrement Amazon S3 n'utilise ni n'interagit avec les clés de compartiment, même si vous spécifiez une clé KMS comme clé d'encapsulation.

Le Manuel du développeur du client de chiffrement Amazon S3 se concentre sur les versions 3.0 et ultérieures du client de chiffrement Amazon S3. Pour plus d'informations, consultez What is the Amazon S3 Encryption Client? (Qu'est-ce que le client de chiffrement Amazon S3) dans le Guide du développeur du client de chiffrement Amazon S3.

Pour plus d'informations sur les versions précédentes du client Amazon S3 Encryption, consultez le guide du développeur du AWS SDK correspondant à votre langage de programmation.

- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for Go
- AWS SDK for PHP
- AWS SDK for Ruby
- AWS SDK for C++

Confidentialité du trafic inter-réseaux

Cette rubrique décrit comment Amazon S3 sécurise les connexions depuis le service vers d'autres emplacements.

Trafic entre les clients de service et sur site et les applications

Les connexions suivantes peuvent être combinées AWS PrivateLink pour fournir une connectivité entre votre réseau privé et AWS :

- Une connexion AWS VPN de site à site. Pour plus d'informations, voir Qu'est-ce que c'est AWS Site-to-Site VPN ?
- Une AWS Direct Connect connexion. Pour plus d'informations, voir Qu'est-ce que c'est AWS Direct Connect ?

L'accès à Amazon S3 via le réseau se fait via des API AWS publiées. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2. Nous recommandons TLS 1.3. Les clients doivent également prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy)

comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes. De plus, vous devez signer les demandes à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète, associées à un principal IAM. Vous pouvez également utiliser le service AWS Security Token Service (STS) afin de générer des informations d'identification de sécurité temporaires pour signer les demandes.

Trafic entre les AWS ressources d'une même région

Un point de terminaison de VPC (Virtual Private Cloud) pour Amazon S3 est une entité logique au sein d'un VPC qui autorise la connectivité uniquement à Amazon S3. Le VPC achemine les demandes vers Amazon S3 et les réponses en retour vers le VPC. Pour plus d'informations, consultez Points de terminaison d'un VPC dans le Guide de l'utilisateur de VPC. Pour obtenir des exemples de stratégie de compartiment que vous pouvez utiliser pour contrôler l'accès aux compartiments S3 à partir de points de terminaison d'un VPC, veuillez consulter Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.

AWS PrivateLink pour Amazon S3

Avec AWS PrivateLink Amazon S3, vous pouvez provisionner des points de terminaison VPC d'interface (points de terminaison d'interface) dans votre cloud privé virtuel (VPC). Ces points de terminaison sont directement accessibles depuis des applications installées sur site via VPN et/ou via un autre Région AWS système de peering VPC. AWS Direct Connect

Les points de terminaison d'interface sont représentés par une ou plusieurs interfaces réseau Elastic (ENI) auxquelles des adresses IP privées sont attribuées à partir de sous-réseaux VPC. Les demandes adressées à Amazon S3 via les points de terminaison d'interface restent sur le réseau Amazon. Vous pouvez également accéder aux points de terminaison de l'interface de votre VPC à partir d'applications sur site AWS Direct Connect via AWS Virtual Private Network ou ().AWS VPN Pour plus d'informations sur la façon de connecter votre VPC à votre réseau sur site, consultez le AWS Direct Connect Guide de l'utilisateur et le AWS Site-to-Site VPN Guide de l'utilisateur .

Pour des informations générales sur les points de terminaison d'interface, consultez Points de terminaison de VPC d'interface (AWS PrivateLink) dans le Guide AWS PrivateLink .

Rubriques

- Types de points de terminaison de VPC pour Amazon S3
- Restrictions et limites de AWS PrivateLink pour Amazon S3

- Création d'un point de terminaison d'un VPC
- Accès aux points de terminaison d'interface d'Amazon S3
- DNS privé
- Accès aux compartiments, aux points d'accès et aux opérations d'API de contrôle Amazon S3 depuis les points de terminaison de l'interface S3
- Mise à jour d'une configuration DNS sur site
- Création d'une stratégie de point de terminaison de VPC pour Amazon S3

Types de points de terminaison de VPC pour Amazon S3

Vous pouvez utiliser deux types de points de terminaison VPC pour accéder à Amazon S3 : les points de terminaison de passerelle et les points de terminaison d'interface (en utilisant). AWS PrivateLink Un point de terminaison de passerelle est une passerelle que vous spécifiez dans votre table de routage pour accéder à Amazon S3 depuis votre VPC via le AWS réseau. Les points de terminaison d'interface étendent les fonctionnalités des points de terminaison de passerelle en utilisant des adresses IP privées pour acheminer les demandes vers Amazon S3 depuis votre VPC, sur site, ou depuis un VPC dans un autre en utilisant Région AWS le peering VPC ou. AWS Transit Gateway Pour plus d'informations, consultez Qu'est-ce que l'appairage de VPC ? et Transit Gateway vs VPC peering (Passerelle de transit ou appairage de VPC).

Les points de terminaison d'interface sont compatibles avec les points de terminaison de passerelle. Si vous avez un point de terminaison de passerelle existant dans votre VPC, vous pouvez utiliser les deux types de points de terminaison dans le même VPC.

Points de terminaison de passerelle pour Amazon S3	Points de terminaison d'interface pour Amazon S3
Dans les deux cas, votre trafic réseau reste sur le AWS réseau.	
Utiliser des adresses IP publiques Amazon S3	Utiliser des adresses IP privées depuis votre VPC pour accéder à Amazon S3
Utiliser les mêmes noms DNS Amazon S3	Exiger des noms DNS Amazon S3 spécifiques aux points de terminaison
N'autorise pas l'accès sur site	Autoriser l'accès depuis vos sites

Points de terminaison de passerelle pour Amazon S3	Points de terminaison d'interface pour Amazon S3
Ne pas autoriser l'accès depuis un autre Région AWS	Autoriser l'accès d'un VPC à un autre en utilisant le Région AWS peering VPC ou AWS Transit Gateway
Non facturé	Facturé

Pour plus d'informations sur les points de terminaison de passerelle, consultez Points de terminaison de VPC de passerelle dans le Guide AWS PrivateLink .

Restrictions et limites de AWS PrivateLink pour Amazon S3

Les limites du VPC s'appliquent AWS PrivateLink à Amazon S3. Pour plus d'informations, consultez Considérations sur les points de terminaison d'interface et Quotas AWS PrivateLink dans le Guide AWS PrivateLink . En outre, les restrictions suivantes s'appliquent.

AWS PrivateLink pour Amazon S3 ne prend pas en charge les éléments suivants :

- Points de terminaison FIPS (Federal Information Processing Standard)
- Points de terminaison de sites web
- Points de terminaison globaux hérités
- Points de terminaison S3-région
- Points de terminaison double pile Amazon S3
- En utilisant CopyObjectou UploadPartCopyentre des seaux dans différents Régions AWS
- Protocole TLS (Transport Layer Security) 1.1

Création d'un point de terminaison d'un VPC

Pour créer un point de terminaison d'interface de VPC, veuillez consulter Création d'un point de terminaison de VPC dans le Guide AWS PrivateLink .

Accès aux points de terminaison d'interface d'Amazon S3

Lorsque vous créez un point de terminaison d'interface, Amazon S3 génère deux types de noms DNS S3 spécifiques au point de terminaison : des noms Régionaux et des noms zonaux.

- Un nom DNS régional inclut un ID de point de terminaison VPC unique, un identifiant de service Région AWS, le et `vpce.amazonaws.com` dans son nom. Par exemple, pour l'ID de point de terminaison de VPC *`vpce-1a2b3c4d`*, le nom DNS généré peut être similaire à *`vpce-1a2b3c4d-5e6f`*`.s3.us-east-1.vpce.amazonaws.com`.
- Les noms DNS zonaux incluent la zone de disponibilité, par exemple, *`vpce-1a2b3c4d-5e6f`*`-`**`us-east-1a.`**`s3.us-east-1.vpce.amazonaws.com`. Vous pouvez utiliser cette option si votre architecture isole les zones de disponibilité. Par exemple, vous pouvez l'utiliser pour contenir les pannes ou réduire les coûts de transfert de données Régionaux.

Les noms DNS S3 spécifiques au point de terminaison peuvent être résolus depuis le domaine DNS public S3.

DNS privé

Les options de DNS privé pour les points de terminaison de l'interface d'un VPC simplifient le routage du trafic S3 vers les points de terminaison d'un VPC et vous aident à tirer parti du chemin réseau le plus économique disponible pour votre application. Vous pouvez utiliser des options de DNS privé pour acheminer le trafic S3 régional sans mettre à jour vos clients S3 pour utiliser les noms DNS spécifiques aux points de terminaison de vos points de terminaison d'interface, ni gérer l'infrastructure DNS. Lorsque les noms DNS privés sont activés, les requêtes DNS S3 régionales sont résolues vers les adresses IP privées AWS PrivateLink des points de terminaison suivants :

- Points de terminaison régionaux de compartiment (par exemple, `s3.us-east-1.amazonaws.com`)
- Points de contrôle (par exemple, `s3-control.us-east-1.amazonaws.com`)
- Points de terminaison des points d'accès (par exemple, `s3-accesspoint.us-east-1.amazonaws.com`)

Si votre VPC comporte un point de terminaison de passerelle, vous pouvez automatiquement acheminer les demandes internes au VPC via votre point de terminaison de passerelle S3 existant et les demandes sur site via le point de terminaison de votre interface. Cette approche vous permet d'optimiser vos coûts de mise en réseau en utilisant des points de terminaison de passerelle, qui ne sont pas facturés, pour votre trafic interne au VPC. Vos applications sur site peuvent être utilisées à l' AWS PrivateLink aide du point de terminaison Resolver entrant. Amazon fournit un serveur DNS pour votre VPC, appelé Route 53 Resolver. Un point de terminaison Resolver entrant réachemine des requêtes DNS à partir du réseau sur site vers Route 53 Resolver.

> **Important**
>
> Pour bénéficier du chemin réseau le plus économique lorsque vous utilisez l'option Activer le DNS privé uniquement pour les points de terminaison entrants, un point de terminaison de passerelle doit être présent dans votre VPC. La présence d'un point de terminaison de passerelle permet de garantir que le trafic interne au VPC passe toujours par le réseau privé d' AWS lorsque l'option Activer le DNS privé uniquement pour les points de terminaison entrants est sélectionnée. Vous devez conserver ce point de terminaison de passerelle tant que l'option Activer le DNS privé uniquement pour les points de terminaison entrants est sélectionnée. Si vous souhaitez supprimer le point de terminaison de votre passerelle, vous devez d'abord désactiver Activer le DNS privé uniquement pour les points de terminaison entrants.
>
> Si vous souhaitez mettre à jour un point de terminaison d'interface existant pour Activer le DNS privé uniquement pour les points de terminaison entrants, vérifiez d'abord que votre VPC possède un point de terminaison de passerelle S3. Pour plus d'informations sur les points de terminaison de passerelle et la gestion des noms DNS privés, consultez Points de terminaison de passerelle et Gestion des noms DNS privés dans le Guide AWS PrivateLink .

L'option Activer le DNS privé uniquement pour les points de terminaison entrants n'est disponible que pour les services qui prennent en charge les points de terminaison de passerelle.

Pour plus d'informations sur la création d'un point de terminaison d'un VPC utilisant l'option Activer le DNS privé uniquement pour les points de terminaison entrants, consultez Créer un point de terminaison d'interface dans le Guide AWS PrivateLink .

Utilisation de la console du VPC

Dans la console, vous avez deux options : Activer le nom DNS et Activer le DNS privé uniquement pour les points de terminaison entrants. Activer le nom DNS est une option prise en charge par AWS PrivateLink. En utilisant l'option Activer le nom DNS, vous pouvez utiliser la connectivité privée d'Amazon à Amazon S3, tout en envoyant des demandes aux noms DNS des points de terminaison publics par défaut. Lorsque cette option est activée, les clients peuvent bénéficier du chemin réseau le plus économique disponible pour leur application.

Lorsque vous activez les noms DNS privés sur un point de terminaison d'interface de VPC existant ou nouveau pour Amazon S3, l'option Activer le DNS privé uniquement pour les points de terminaison entrants est sélectionnée par défaut. Si cette option est sélectionnée, vos applications utilisent

uniquement des points de terminaison d'interface pour votre trafic sur site. Ce trafic interne au VPC utilise automatiquement les points de terminaison de passerelle les moins coûteux. Vous pouvez également désactiver l'option Activer le DNS privé uniquement pour les points de terminaison entrants afin d'acheminer toutes les demandes S3 via le point de terminaison de votre interface.

En utilisant le AWS CLI

Si vous ne spécifiez pas de valeur pour `PrivateDnsOnlyForInboundResolverEndpoint`, la valeur par défaut est `true`. Toutefois, avant que votre VPC n'applique vos paramètres, il vérifie qu'un point de terminaison de passerelle est présent dans le VPC. Si un point de terminaison de passerelle est présent dans le VPC, l'appel aboutit. Sinon, le message d'erreur suivant s'affiche :

Pour que ce paramètre PrivateDnsOnlyForInboundResolverEndpoint soit défini sur true, le VPC *`vpce_id`* doit disposer d'un point de terminaison de passerelle pour le service.

Pour un nouveau point de terminaison d'interface de VPC

Utilisez les attributs `private-dns-enabled` et `dns-options` pour activer le DNS privé via la ligne de commande. L'option `PrivateDnsOnlyForInboundResolverEndpoint` de l'attribut `dns-options` doit être définie sur `true`. Remplacez *`user input placeholders`* par vos propres informations.

```
aws ec2 create-vpc-endpoint \
--region us-east-1 \
--service-name s3-service-name \
--vpc-id client-vpc-id \
--subnet-ids client-subnet-id \
--vpc-endpoint-type Interface  \
--private-dns-enabled  \
--ip-address-type ip-address-type \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true \
--security-group-ids client-sg-id
```

Pour un point de terminaison d'un VPC existant

Si vous souhaitez utiliser un DNS privé pour un point de terminaison d'un VPC existant, utilisez l'exemple de commande suivant et remplacez *`user input placeholders`* par vos propres informations.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

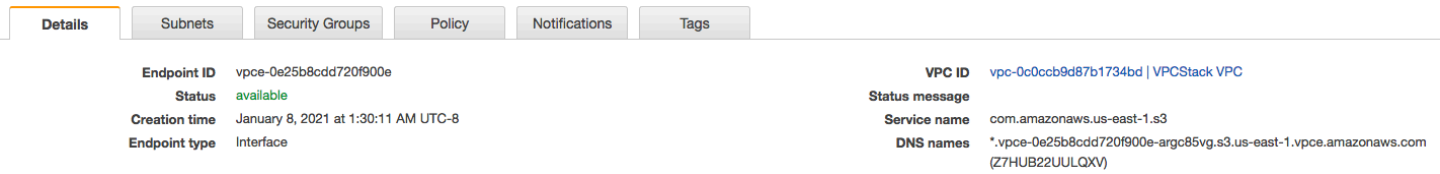
Si vous souhaitez mettre à jour un point de terminaison d'un VPC existant afin d'activer le DNS privé uniquement pour le résolveur entrant, utilisez l'exemple suivant et remplacez les exemples de valeurs par les vôtres.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Accès aux compartiments, aux points d'accès et aux opérations d'API de contrôle Amazon S3 depuis les points de terminaison de l'interface S3

Vous pouvez utiliser les AWS SDK AWS CLI ou pour accéder aux compartiments, aux points d'accès S3 et aux opérations de l'API Amazon S3 Control via les points de terminaison de l'interface S3.

L'image suivante illustre l'onglet Détails de la console VPC, où vous pouvez trouver le nom DNS d'un point de terminaison de VPC. Dans cet exemple, l'ID de point de terminaison de VPC (vpce-id) est `vpce-0e25b8cdd720f900e` et le nom DNS est `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`.



Lorsque vous utilisez le nom DNS pour accéder à une ressource, remplacez * par la valeur appropriée. Les valeurs appropriées à utiliser à la place de * sont les suivantes :

- `bucket`

- `accesspoint`
- `control`

Par exemple, pour accéder à un compartiment, utilisez un nom DNS comme celui-ci :

`bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`

Pour obtenir des exemples d'utilisation des noms DNS pour accéder aux compartiments, aux points d'accès et aux opérations d'API de contrôle Amazon S3, consultez les sections suivantes de AWS CLI exemples et AWS Exemples de SDK.

Pour plus d'informations sur la façon d'afficher vos noms DNS spécifiques aux terminaux, consultez Viewing endpoint service private DNS name configuration (Affichage de la configuration du nom DNS privé du service de point de terminaison) dans le Guide de l'utilisateur VPC.

AWS CLI exemples

Pour accéder aux compartiments S3, aux points d'accès S3 ou aux opérations de l'API Amazon S3 Control via les points de terminaison de l'interface S3 dans AWS CLI les commandes, utilisez les paramètres `--region` et`--endpoint-url`.

Exemple : utilisez une URL de point de terminaison pour répertorier les objets dans votre compartiment

Dans l'exemple suivant, remplacez le nom du compartiment *`my-bucket`*, la région *`us-east-1`* et le nom DNS de l'ID du point de terminaison de VPC *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* par vos propres informations.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
 https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Exemple : utilisez une URL de point de terminaison pour répertorier les objets depuis un point d'accès

- Méthode 1 : utilisation de l'Amazon Resource Name (ARN) du point d'accès avec le point de terminaison du point d'accès

  Remplacez l'ARN *`us-east-1:123456789012:accesspoint/accesspointexamplename`*, la région *`us-east-1`* et l'ID du point de terminaison d'un VPC *`vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`* par vos propres informations.

```
aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/
accesspointexamplename --region us-east-1 --endpoint-url
 https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Si vous ne parvenez pas à exécuter correctement la commande, AWS CLI installez la dernière version et réessayez. Pour plus d'informations sur les instructions de mise à jour, consultez Installation ou mise à jour de la dernière version de l'interface AWS CLI dans le Guide de l'utilisateur AWS Command Line Interface .

- Méthode 2 : utilisation de l'alias du point d'accès avec le point de terminaison régional du compartiment

  Dans l'exemple suivant, remplacez l'alias du point d'accès *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la région *us-east-1* et l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

  ```
  aws s3api list-objects-v2 --
  bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias
   --region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
  east-1.vpce.amazonaws.com
  ```

- Méthode 3 : utilisation de l'alias du point d'accès avec le point de terminaison du point d'accès

  Tout d'abord, pour construire un point de terminaison S3 avec le compartiment inclus dans le nom d'hôte, définissez le style d'adressage sur `virtual` pour `aws s3api`. Pour plus d'informations sur `AWS configure`, consultez Configuration and credential file settings (Paramètres des fichiers de configuration et d'informations d'identification) dans le Guide de l'utilisateur AWS Command Line Interface .

  ```
  aws configure set default.s3.addressing_style virtual
  ```

  Ensuite, dans l'exemple suivant, remplacez l'alias du point d'accès *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, la région *us-east-1* et l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations. Pour plus d'informations sur l'alias du point d'accès, consultezUtilisation d'un alias de type compartiment pour votre point d'accès de compartiment S3.

```
aws s3api list-objects-v2 --
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias --
region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

Exemple : utilisez une URL de point de terminaison pour répertorier les tâches avec une opération d'API de contrôle S3

Dans l'exemple suivant, remplacez la région *us-east-1*, l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et l'ID du compte *12345678* par vos propres informations.

```
aws s3control --region us-east-1 --endpoint-url
 https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --
account-id 12345678
```

AWS Exemples de SDK

Pour accéder aux compartiments S3, aux points d'accès S3 ou aux opérations de l'API Amazon S3 Control via les points de terminaison de l'interface S3 lorsque vous utilisez les AWS SDK, mettez à jour vos SDK avec la dernière version. Configurez ensuite vos clients pour qu'ils utilisent une URL de point de terminaison afin d'accéder à un compartiment, un point d'accès ou des opérations d'API de contrôle Amazon S3 via des points de terminaison d'interface S3.

SDK for Python (Boto3)

Exemple : Utiliser une URL de point de terminaison pour accéder à un compartiment S3

Dans l'exemple suivant, remplacez la région *us-east-1* et l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

```
s3_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

Exemple : Utiliser une URL de point de terminaison pour accéder à un point d'accès S3

Dans l'exemple suivant, remplacez la région *us-east-1* et l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

```
ap_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com'
)
```

Exemple : utilisez une URL de point de terminaison pour accéder à l'API de contrôle Amazon S3

Dans l'exemple suivant, remplacez la région *us-east-1* et l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

```
control_client = session.client(
service_name='s3control',
region_name='us-east-1',
endpoint_url='https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

SDK for Java 1.x

Exemple : Utiliser une URL de point de terminaison pour accéder à un compartiment S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

```
// bucket client
final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
List<Bucket> buckets = s3.listBuckets();
```

Exemple : Utiliser une URL de point de terminaison pour accéder à un point d'accès S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et l'ARN *us-east-1:123456789012:accesspoint/prod* par vos propres informations.

```
// accesspoint client
final AmazonS3 s3accesspoint =
 AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
ObjectListing objects = s3accesspoint.listObjects("arn:aws:s3:us-
east-1:123456789012:accesspoint/prod");
```

Exemple : utilisez une URL de point de terminaison pour accéder à une opération d'API de contrôle Amazon S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* par vos propres informations.

```
// control client
final AWSS3Control s3control =
 AWSS3ControlClient.builder().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
final ListJobsResult jobs = s3control.listJobs(new ListJobsRequest());
```

SDK for Java 2.x

Exemple : Utiliser une URL de point de terminaison pour accéder à un compartiment S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et la région *Region.US_EAST_1* par vos propres informations.

```
// bucket client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)
```

```
 .endpointOverride(URI.create("https://bucket.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
                    .build()
```

Exemple : Utiliser une URL de point de terminaison pour accéder à un point d'accès S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et la région *Region.US_EAST_1* par vos propres informations.

```
// accesspoint client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

 .endpointOverride(URI.create("https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
                    .build()
```

Exemple : utilisez une URL de point de terminaison pour accéder à l'API de contrôle Amazon S3

Dans l'exemple suivant, remplacez l'ID du point de terminaison d'un VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et la région *Region.US_EAST_1* par vos propres informations.

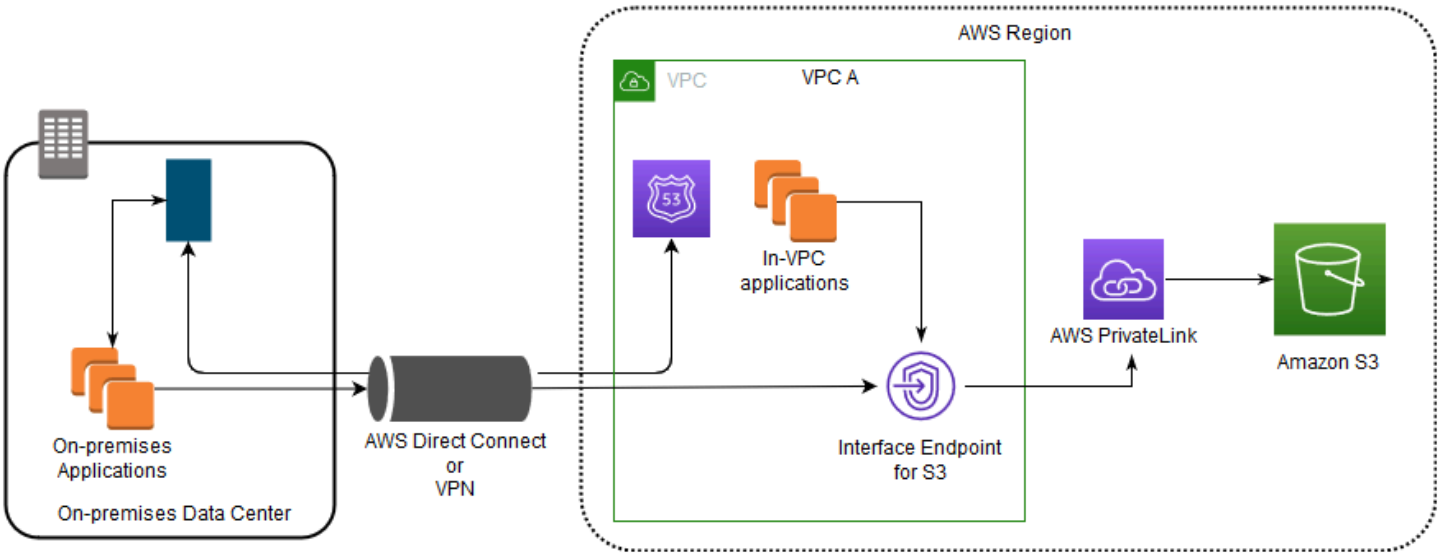
```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com"))
                                    .build()
```

Mise à jour d'une configuration DNS sur site

Lorsque vous utilisez des noms DNS spécifiques aux points de terminaison pour accéder aux points de terminaison d'interface pour Amazon S3, vous n'avez pas besoin de mettre à jour votre résolveur DNS sur site. Vous pouvez résoudre le nom DNS spécifique au point de terminaison avec l'adresse IP privée du point de terminaison d'interface depuis le domaine DNS public Amazon S3.

Utilisation de points de terminaison d'interface pour accéder à Amazon S3 sans point de terminaison de passerelle ou passerelle Internet dans le VPC

Les points de terminaison d'interface de votre VPC peuvent acheminer les applications de VPC et les applications sur site vers Amazon S3 via le réseau Amazon, comme illustré dans le diagramme suivant.



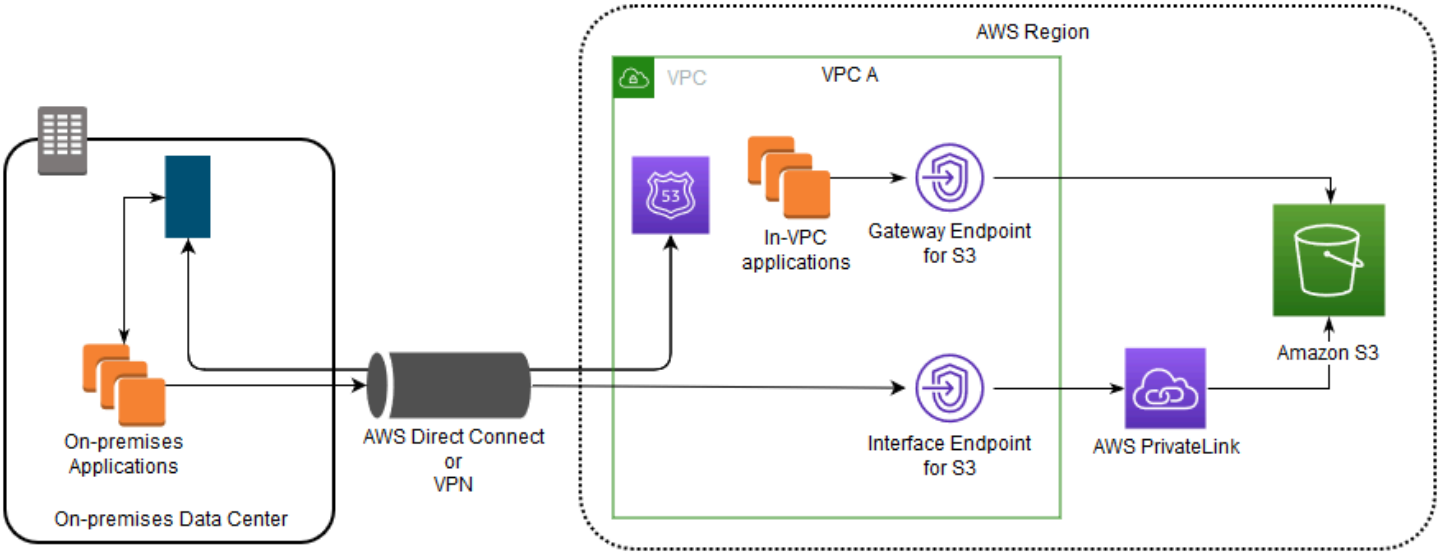
Le diagramme illustre les éléments suivants :

- Votre réseau local utilise AWS Direct Connect ou AWS VPN pour se connecter au VPC A.
- Vos applications sur site et dans le VPC A utilisent des noms DNS spécifiques aux points de terminaison pour accéder à Amazon S3 via le point de terminaison d'interface S3.
- Les applications locales envoient des données au point de terminaison de l'interface dans le VPC AWS Direct Connect via (ou) AWS VPN. AWS PrivateLink déplace les données du point de terminaison de l'interface vers Amazon S3 via le AWS réseau.
- Les applications in-VPC envoient également du trafic vers le point de terminaison de l'interface. AWS PrivateLink déplace les données du point de terminaison de l'interface vers Amazon S3 via le AWS réseau.

Utilisation conjointe de points de terminaison de passerelle et de points de terminaison d'interface dans le même VPC pour accéder à Amazon S3

Vous pouvez créer des points de terminaison d'interface et conserver le point de terminaison de passerelle existant dans le même VPC, comme le montre le diagramme suivant. En adoptant cette

approche, vous autorisez les applications internes au VPC à continuer d'accéder à Amazon S3 via le point de terminaison de la passerelle, ce qui n'est pas facturé. Ensuite, seules vos applications sur site utilisent des points de terminaison d'interface pour accéder à Amazon S3. Pour accéder à Amazon S3 de cette façon, vous devez mettre à jour vos applications sur site afin d'utiliser des noms DNS spécifiques du point de terminaison pour Amazon S3.



Le diagramme illustre les éléments suivants :

- Les applications locales utilisent des noms DNS spécifiques au point de terminaison pour envoyer des données au point de terminaison de l'interface au sein du VPC via (ou). AWS Direct Connect AWS VPN AWS PrivateLink déplace les données du point de terminaison de l'interface vers Amazon S3 via le AWS réseau.
- À l'aide des noms régionaux Amazon S3 par défaut, les applications intégrées au VPC envoient des données au point de terminaison de la passerelle qui se connecte à Amazon S3 via le AWS réseau.

Pour plus d'informations sur les points de terminaison de passerelle, consultez la sectionPoints de terminaison d'un VPC de passerelle du Guide de l'utilisateur VPC.

Création d'une stratégie de point de terminaison de VPC pour Amazon S3

Vous pouvez attacher une stratégie de point de terminaison à votre point de terminaison de VPC qui contrôle l'accès à Amazon S3. La stratégie spécifie les informations suivantes :

- Le principal AWS Identity and Access Management (IAM) qui peut effectuer des actions
- Les actions qui peuvent être effectuées.

- Les ressources sur lesquelles les actions peuvent être exécutées.

Vous pouvez également utiliser des stratégies de compartiment Amazon S3 pour restreindre l'accès à des compartiments spécifiques depuis un point de terminaison d'un VPC spécifique en utilisant la condition `aws:sourceVpce` de votre stratégie de compartiment. Les exemples suivants montrent les stratégies qui restreignent l'accès à un compartiment ou à un point de terminaison.

Rubriques

Exemple : restriction de l'accès à un compartiment spécifique depuis le point de terminaison d'un VPC

Vous pouvez créer une stratégie de point de terminaison qui restreint l'accès à des compartiments Amazon S3 spécifiques uniquement. Ce type de politique est utile si d'autres Services AWS politiques de votre VPC utilisent des buckets. La stratégie de compartiment suivante restreint l'accès à *`example-s3-bucket1`* uniquement. Pour utiliser cette politique de point de terminaison, remplacez *`example-s3-bucket1`* par le nom de votre compartiment.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::example-s3-bucket1",
                   "arn:aws:s3:::example-s3-bucket1/*"]
    }
```

```
  ]
}
```

Exemple : restriction de l'accès aux compartiments dans un compte spécifique depuis le point de terminaison d'un VPC

Vous pouvez créer une politique de point de terminaison qui restreint l'accès aux compartiments S3 d'un point spécifique. Compte AWS Pour empêcher les clients de votre VPC d'accéder aux compartiments dont vous n'êtes pas le propriétaire, utilisez la déclaration suivante dans votre stratégie de point de terminaison. L'exemple de déclaration suivant crée une stratégie qui restreint l'accès aux ressources appartenant à un seul ID de Compte AWS , *111122223333*.

```
{
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

> **Note**
>
> Pour spécifier l' Compte AWS ID de la ressource à laquelle vous accédez, vous pouvez utiliser la clé `aws:ResourceAccount` ou la `s3:ResourceAccount` clé dans votre politique IAM. Sachez toutefois que certains Services AWS dépendent de l'accès à des buckets AWS gérés. Par conséquent, l'utilisation de la clé `aws:ResourceAccount` ou `s3:ResourceAccount` dans votre politique IAM risque également d'avoir un impact sur l'accès à ces ressources.

Exemple : restriction de l'accès au point de terminaison d'un VPC spécifique dans la stratégie de compartiment S3

Exemple : restriction de l'accès au point de terminaison d'un VPC spécifique dans la stratégie de compartiment S3

La stratégie de compartiment Amazon S3 suivante autorise l'accès à un compartiment spécifique, *example-s3-bucket2*, depuis le point de terminaison d'un VPC *vpce-1a2b3c4d* uniquement. La politique refuse tout accès au compartiment si le point de terminaison spécifié n'est pas utilisé. La condition `aws:sourceVpce` spécifie le point de terminaison et ne requiert pas d'Amazon Resource Name (ARN) pour la ressource de point de terminaison d'un VPC, mais uniquement l'ID du point de terminaison. Pour utiliser cette stratégie de point de terminaison, remplacez *example-s3-bucket2* et *vpce-1a2b3c4d* par le nom et le point de terminaison de votre compartiment.

> **Important**
>
> - Lors de l'application de la stratégie de compartiment Amazon S3 suivante afin de restreindre l'accès à certains points de terminaison d'un VPC uniquement, vous pouvez bloquer involontairement votre accès au compartiment. Les stratégies de compartiment dans le but de restreindre l'accès aux connexions issues du point de terminaison de votre VPC peuvent bloquer toutes les connexions à ce compartiment. Pour des informations sur la correction de ce problème, veuillez consulter Ma politique de compartiment n'a pas le bon VPC ou ID de point de terminaison d'un VPC. Comment puis-je corriger la politique de façon à pouvoir accéder au compartiment ? que vous trouverez dans le AWS Support Centre de connaissances.
> - Avant d'utiliser l'exemple de stratégie suivant, remplacez l'ID de point de terminaison du VPC par une valeur appropriée pour votre cas d'utilisation. Dans le cas contraire, vous ne parviendrez pas à accéder à votre compartiment.
> - Cette stratégie désactive l'accès à la console du compartiment spécifié, car les demandes de la console ne proviennent pas du point de terminaison du VPC défini.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
```

```
    { "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::example-s3-bucket2",
                   "arn:aws:s3:::example-s3-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Pour plus d'exemples de stratégie, consultez la section Points de terminaison pour Amazon S3 du Guide de l'utilisateur VPC.

Pour plus d'informations sur la connectivité VPC, consultez la section Options de connectivité réseau à VPC dans le livre blanc Amazon AWS Virtual Private Cloud Connectivity Options.

Gestion des accès

Dans AWS, une ressource est une entité avec laquelle vous pouvez travailler. Dans Amazon Simple Storage Service (S3), les compartiments et les objets sont les ressources Amazon S3 d'origine. Chaque client S3 possède probablement des compartiments contenant des objets. Au fur et à mesure que de nouvelles fonctionnalités ont été ajoutées à S3, des ressources supplémentaires ont également été ajoutées, mais tous les clients n'utilisent pas ces ressources spécifiques aux fonctionnalités. Pour plus d'informations sur les ressources Amazon S3, consultezRessources S3.

Par défaut, toutes les ressources Amazon S3 sont privées. Par défaut, l'utilisateur root Compte AWS qui a créé la ressource (propriétaire de la ressource) et les utilisateurs IAM de ce compte disposant des autorisations nécessaires peuvent accéder à une ressource qu'ils ont créée. Le propriétaire de la ressource décide qui d'autre peut accéder à la ressource et les actions que les autres sont autorisés à effectuer sur la ressource. S3 dispose de divers outils de gestion des accès que vous pouvez utiliser pour accorder à d'autres personnes l'accès à vos ressources S3.

Les sections suivantes présentent un aperçu des ressources S3, des outils de gestion des accès S3 disponibles et des meilleurs cas d'utilisation pour chaque outil de gestion des accès. Les listes de ces sections visent à être exhaustives et incluent toutes les ressources S3, les outils de gestion des accès et les cas d'utilisation courants de la gestion des accès. Dans le même temps, ces sections sont conçues pour être des répertoires qui vous mènent aux détails techniques que vous souhaitez. Si vous comprenez bien certains des sujets suivants, vous pouvez passer directement à la section qui vous concerne.

Rubriques

Ressources S3

Les ressources Amazon S3 d'origine sont les compartiments et les objets qu'ils contiennent. Au fur et à mesure que de nouvelles fonctionnalités sont ajoutées à S3, de nouvelles ressources sont également ajoutées. Vous trouverez ci-dessous une liste complète des ressources S3 et de leurs fonctionnalités respectives.

Type de ressource	Fonctionnalité Amazon S3	Description
`bucket`	Fonctions de base	Un compartiment est un conteneur d'objets. Pour stocker un objet dans S3, créez un compartiment, puis chargez un ou plusieurs objets dans le compartiment. Pour plus d’informations, consultez Création, configuration et utilisation des compartiments Amazon S3.
`object`		Un objet peut être un fichier et toutes les métadonnées décrivant ce fichier. Lorsqu'un objet se trouve dans le

Type de ressource	Fonctionnalité Amazon S3	Description
		compartiment, vous pouvez l'ouvrir, le télécharger et le déplacer. Pour plus d'informations, consultez Chargement, téléchargement et utilisation des objets dans Amazon S3.
`accesspoint`	Points d'accès	Les points d'accès sont appelés points de terminaison réseau attachés à des compartiments que vous pouvez utiliser pour effectuer des opérations sur des objets Amazon S3, telles que `GetObject` et. `PutObject` Chaque point d'accès dispose d'autorisations, de contrôles réseau distincts et d'une politique de point d'accès personnalisée qui fonctionne conjointement avec la politique de compartiment attachée au compartiment sous-jacent. Vous pouvez configurer n'importe quel point d'accès pour accepter uniquement les demandes provenant d'un cloud privé virtuel (VPC) ou configurer des paramètres de blocage d'accès public personnalisés pour chaque point d'accès. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.
`objectlambdaaccesspoint`		Un point d'accès Object Lambda est un point d'accès pour un bucket qui est également associé à une fonction Lambda. Avec Object Lambda Access Point, vous pouvez ajouter votre propre code à Amazon S3 et `GET` demander `LIST` à modifier et `HEAD` à traiter les données lorsqu'elles sont renvoyées à une application. Pour plus d'informations, consultez Création de points d'accès Object Lambda.

Type de ressource	Fonctionnalité Amazon S3	Description
`multiregionaccesspoint`		Les points d'accès multirégionaux fournissent un point de terminaison global que les applications peuvent utiliser pour traiter les demandes provenant de compartiments Amazon S3 situés dans plusieurs AWS régions. Vous pouvez utiliser des points d'accès multi-régions pour créer des applications multi-régions avec la même architecture utilisée dans une seule région, puis exécuter ces applications partout dans le monde. Au lieu d'envoyer des demandes via l'Internet public congestionné, les demandes d'application adressées à un point d'accès global multirégional sont automatiquement acheminées via le réseau AWS mondial vers le compartiment Amazon S3 le plus proche. Pour plus d'informations, consultez Points d'accès multi-régions dans Amazon S3.
`job`	Opérations par lot S3	Une tâche est une ressource de la fonctionnalité S3 Batch Operations. Vous pouvez utiliser S3 Batch Operations pour effectuer des opérations par lots à grande échelle sur des listes d'objets Amazon S3 que vous spécifiez. Amazon S3 suit la progression de la tâche d'opération par lots, envoie des notifications et stocke un rapport d'achèvement détaillé de toutes les actions, vous offrant ainsi une expérience entièrement gérée, auditable et sans serveur. Pour plus d'informations, consultez Exécution des opérations par lot à grande échelle sur des objets Amazon S3.

Type de ressource	Fonctionnalité Amazon S3	Description
`storagelensconfiguration`	S3 Storage Lens	Une configuration S3 Storage Lens collecte des métriques de stockage à l'échelle de l'organisation et des données utilisateur sur tous les comptes. S3 Storage Lens fournit aux administrateurs une vue unique de l'utilisation et de l'activité du stockage d'objets sur des centaines, voire des milliers de comptes au sein d'une organisation, avec des informations détaillées permettant de générer des informations à plusieurs niveaux d'agrégation. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec Amazon S3 Storage Lens.
`storagelensgroup`		Un groupe S3 Storage Lens agrège les métriques à l'aide de filtres personnalisés basés sur les métadonnées des objets. Les groupes S3 Storage Lens vous aident à étudier les caractéristiques de vos données, telles que la répartition des objets par âge, les types de fichiers les plus courants, etc. Pour plus d'informations, consultez Utilisation des groupes S3 Storage Lens.
`accessgrantsinstance`	Octrois d'accès S3	Une instance S3 Access Grants est un conteneur pour les autorisations S3 que vous créez. Avec S3 Access Grants, vous pouvez créer des autorisations pour vos données Amazon S3 pour les identités IAM au sein de votre compte, les identités IAM sur d'autres comptes (comptes intercomptes) et les identités de répertoire ajoutées à AWS IAM Identity Center partir de votre annuaire d'entreprise. Pour plus d'informations sur les subventions d'accès S3, consultezGestion de l'accès avec les octrois d'accès S3.

Type de ressource	Fonctionnalité Amazon S3	Description
`accessgrantslocation`		Un emplacement d'autorisations d'accès est un compartiment, un préfixe dans un compartiment ou un objet que vous enregistrez dans votre instance S3 Access Grants. Vous devez enregistrer des emplacements au sein de l'instance S3 Access Grants avant de pouvoir créer une autorisation pour cet emplacement. Ensuite, avec S3 Access Grants, vous pouvez accorder l'accès au compartiment, au préfixe ou à l'objet pour les identités IAM de votre compte, les identités IAM d'autres comptes (comptes multiples) et les identités de répertoire ajoutées à AWS IAM Identity Center partir de votre annuaire d'entreprise. Pour plus d'informations sur les subventions d'accès S3, voir Gestion de l’accès avec les octrois d’accès S3
`accessgrant`		Une autorisation d'accès est une autorisation individuelle accordée à vos données Amazon S3. Avec S3 Access Grants, vous pouvez créer des autorisations pour vos données Amazon S3 pour les identités IAM au sein de votre compte, les identités IAM sur d'autres comptes (comptes intercomptes) et les identités de répertoire ajoutées à AWS IAM Identity Center partir de votre annuaire d'entreprise. Pour plus d'informations sur les subventions d'accès S3, voir Gestion de l’accès avec les octrois d’accès S3

Compartiments

Il existe deux types de compartiments Amazon S3 : les compartiments à usage général et les compartiments de répertoire.

- Les compartiments à usage général constituent le type de compartiment S3 d’origine et sont recommandés pour la plupart des cas d’utilisation et des modèles d’accès. Les compartiments à usage général autorisent également les objets stockés dans toutes les classes de stockage, à l’exception de S3 Express One Zone. Pour plus d'informations sur les classes de stockage S3, consultezUtilisation des classes de stockage Simple Storage Service (Amazon S3).

- Les compartiments d'annuaire utilisent la classe de stockage S3 Express One Zone, recommandée si votre application est sensible aux performances et bénéficie d'une milliseconde et de latences à un chiffre. PUT GET Pour plus d'informations, consultez Compartiments de répertoire, Qu'est-ce que S3 Express One Zone ? et AWS Identity and Access Management (IAM) pour S3 Express One Zone.

Catégorisation des ressources S3

Amazon S3 fournit des fonctionnalités permettant de classer et d'organiser vos ressources S3. La catégorisation de vos ressources est non seulement utile pour les organiser, mais vous pouvez également définir des règles de gestion des accès en fonction des catégories de ressources. En particulier, les préfixes et le balisage sont deux fonctionnalités d'organisation du stockage que vous pouvez utiliser lors de la définition des autorisations de gestion des accès.

> **Note**
>
> Les informations suivantes s'appliquent aux seaux à usage général. Les compartiments de répertoire ne prennent pas en charge le balisage et leur nombre de préfixes est limité. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

- Préfixes : dans Amazon S3, un préfixe est une chaîne de caractères située au début du nom d'une clé d'objet utilisée pour organiser les objets stockés dans vos compartiments S3. Vous pouvez utiliser un caractère séparateur, tel qu'une barre oblique (/), pour indiquer la fin du préfixe dans le nom de la clé de l'objet. Par exemple, vous pouvez avoir des noms de clé d'objet qui commencent par le `engineering/` préfixe ou des noms de clé d'objet qui commencent par le `marketing/campaigns/` préfixe. L'utilisation d'un délimiteur à la fin de votre préfixe, tel qu'une barre oblique, `/` émule les conventions de dénomination des dossiers et des fichiers. Cependant, dans S3, le préfixe fait partie du nom de la clé de l'objet. Dans les compartiments S3 à usage général, il n'existe pas de hiérarchie de dossiers réelle.

  Amazon S3 permet d'organiser et de regrouper des objets à l'aide de leurs préfixes. Vous pouvez également gérer l'accès aux objets à l'aide de leurs préfixes. Par exemple, vous pouvez limiter l'accès aux seuls objets dont le nom commence par un préfixe spécifique.

  Pour plus d'informations, consultez Organisation des objets à l'aide de préfixes. La console S3 utilise le concept de dossiers, qui, dans les compartiments à usage général, sont essentiellement

des préfixes ajoutés au nom de la clé de l'objet. Pour plus d'informations, consultez Organisation des objets dans la console Amazon S3 à l'aide de dossiers.

- Balises : chaque balise est une paire clé-valeur que vous attribuez aux ressources. Par exemple, vous pouvez étiqueter certaines ressources à l'aide de cette balise`topicCategory=engineering`. Vous pouvez utiliser le balisage pour faciliter la répartition des coûts, la catégorisation et l'organisation, ainsi que le contrôle d'accès. Le marquage des compartiments est uniquement utilisé pour la répartition des coûts. Vous pouvez étiqueter des objets, des objectifs de stockage S3, des tâches et des autorisations d'accès S3 à des fins d'organisation ou de contrôle d'accès. Dans S3 Access Grants, vous pouvez également utiliser le balisage pour la répartition des coûts. À titre d'exemple de contrôle de l'accès aux ressources à l'aide de leurs balises, vous pouvez partager uniquement les objets dotés d'une balise spécifique ou d'une combinaison de balises.

  Pour plus d'informations, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de balises de ressources dans le guide de l'utilisateur IAM.

Identités

Dans Amazon S3, le propriétaire de la ressource est l'identité qui a créé la ressource, telle qu'un bucket ou un objet. Par défaut, seul l'utilisateur root du compte qui a créé la ressource et les identités IAM du compte disposant de l'autorisation requise peuvent accéder à la ressource S3. Les propriétaires de ressources peuvent autoriser d'autres identités à accéder à leurs ressources S3.

Les identités qui ne sont pas propriétaires d'une ressource peuvent demander l'accès à cette ressource. Les demandes adressées à une ressource sont authentifiées ou non authentifiées. Les demandes authentifiées doivent inclure une valeur de signature qui authentifie l'expéditeur de la demande, mais les demandes non authentifiées ne nécessitent pas de signature. Nous vous recommandons de n'accorder l'accès qu'aux utilisateurs authentifiés. Pour plus d'informations sur l'authentification des demandes, veuillez consulter Demandes.

> **Important**
>
> Nous vous recommandons de ne pas utiliser les informations d'identification de l'utilisateur Compte AWS root pour effectuer des demandes authentifiées. Il est préférable de créer un rôle IAM, puis de lui accorder un accès total. Nous appelons les utilisateurs possédant ce rôle des administrateurs. Vous pouvez utiliser les informations d'identification attribuées au rôle d'administrateur, au lieu des informations d'identification de l'utilisateur Compte AWS

root, pour interagir avec AWS et effectuer des tâches, telles que créer un bucket, créer des utilisateurs et accorder des autorisations. Pour plus d'informations, consultez les informations d'identification de l'utilisateur Compte AWS root et les informations d'identification de l'utilisateur IAM dans le Références générales AWS, et consultez les meilleures pratiques de sécurité dans IAM dans le guide de l'utilisateur IAM.

Les identités qui accèdent à vos données dans Amazon S3 peuvent être l'une des suivantes :

Compte AWS owner

Celui Compte AWS qui a créé la ressource. Par exemple, le compte qui a créé le bucket. Ce compte possède la ressource. Pour plus d'informations, consultez la section Utilisateur root du AWS compte.

Identités IAM dans le même compte que le propriétaire Compte AWS

Lors de la configuration de comptes pour les nouveaux membres de l'équipe qui ont besoin d'un accès S3, le Compte AWS propriétaire peut utiliser AWS Identity and Access Management (IAM) pour créer des utilisateurs, des groupes et des rôles. Le Compte AWS propriétaire peut ensuite partager des ressources avec ces identités IAM. Le propriétaire du compte peut également spécifier les autorisations à attribuer aux identités IAM, qui autorisent ou refusent les actions pouvant être effectuées sur les ressources partagées.

Les identités IAM offrent des fonctionnalités accrues, notamment la possibilité de demander aux utilisateurs de saisir des informations de connexion avant d'accéder aux ressources partagées. En utilisant les identités IAM, vous pouvez mettre en œuvre une forme d'authentification multifactorielle IAM (MFA) afin de renforcer la base de votre identité. L'une des meilleures pratiques IAM consiste à créer des rôles pour la gestion des accès au lieu d'accorder des autorisations à chaque utilisateur individuel. Vous attribuez le rôle approprié à chaque utilisateur. Pour plus d’informations, consultez Bonnes pratiques de sécurité dans IAM.

Autres titulaires de AWS comptes et leurs identités IAM (accès entre comptes)

Le Compte AWS propriétaire peut également donner accès aux ressources à d'autres propriétaires de AWS comptes, ou à des identités IAM appartenant à un autre AWS compte.

> **Note**
>
> Délégation d'autorisations — Si une personne Compte AWS possède une ressource, elle peut accorder ces autorisations à une autre personne Compte AWS. Ce compte peut

ensuite déléguer ces autorisations, ou un sous-ensemble d'entre elles, aux utilisateurs du même compte. Cela s'appelle une délégation d'autorisations. Mais un compte qui reçoit des autorisations d'un autre compte ne peut pas déléguer ces autorisations « entre comptes » à un autre Compte AWS.

Utilisateurs anonymes (accès public)

Le Compte AWS propriétaire peut rendre les ressources publiques. Rendre une ressource publique partage techniquement la ressource avec l'utilisateur anonyme. Les compartiments créés depuis avril 2023 bloquent tout accès public par défaut, sauf si vous modifiez ce paramètre. Nous vous recommandons de configurer vos compartiments de manière à bloquer l'accès public et de n'accorder l'accès qu'aux utilisateurs authentifiés. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

Services AWS

Le propriétaire de la ressource peut accorder à un autre AWS service l'accès à une ressource Amazon S3. Par exemple, vous pouvez accorder au AWS CloudTrail service l'`s3:PutObject`autorisation d'écrire des fichiers journaux dans votre compartiment. Pour plus d'informations, consultez la section Fournir l'accès à un AWS service.

Identités des annuaires d'entreprise

Le propriétaire de la ressource peut accorder à des utilisateurs ou à des rôles de votre annuaire d'entreprise l'accès à une ressource S3 à l'aide de S3 Access Grants. Pour plus d'informations sur l'ajout de votre annuaire d'entreprise à AWS IAM Identity Center, voir Qu'est-ce qu'IAM Identity Center ? .

Propriétaires du bucket ou de la ressource

Celui Compte AWS que vous utilisez pour créer des buckets et télécharger des objets possède ces ressources. Le propriétaire d'un bucket peut accorder des autorisations entre comptes à un autre Compte AWS (ou à des utilisateurs d'un autre compte) pour le téléchargement d'objets.

Lorsqu'un propriétaire de compartiment autorise un autre compte à télécharger des objets dans un compartiment, le propriétaire du compartiment est propriétaire par défaut de tous les objets chargés dans son compartiment. Toutefois, si les paramètres de bucket appliqués par le propriétaire du bucket et les paramètres préférés du propriétaire du bucket sont désactivés, le Compte AWS

responsable du téléchargement des objets est propriétaire de ces objets, et le propriétaire du bucket n'a aucune autorisation sur les objets appartenant à un autre compte, avec les exceptions suivantes :

- Le propriétaire du compartiment paie les factures. Le propriétaire du compartiment peut refuser l'accès aux objets ou supprimer des objets dans le compartiment, quel que soit le propriétaire de ces derniers.
- Le propriétaire du bucket peut archiver n'importe quel objet ou restaurer des objets archivés, quel que soit son propriétaire. L'archivage fait référence à la classe de stockage utilisée pour stocker les objets. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.

Outils de gestion des accès

Amazon S3 fournit plusieurs fonctionnalités et outils de sécurité. Vous trouverez ci-dessous une liste complète de ces fonctionnalités et outils. Vous n'avez pas besoin de tous ces outils de gestion des accès, mais vous devez en utiliser un ou plusieurs pour accorder l'accès à vos ressources Amazon S3. L'application appropriée de ces outils peut contribuer à garantir que vos ressources ne sont accessibles qu'aux utilisateurs visés.

L'outil de gestion des accès le plus couramment utilisé est une politique d'accès. Une politique d'accès peut être une politique basée sur les ressources attachée à une AWS ressource, telle qu'une politique de compartiment pour un compartiment. Une politique d'accès peut également être une politique basée sur l'identité attachée à une identité AWS Identity and Access Management (IAM), telle qu'un utilisateur, un groupe ou un rôle IAM. Rédigez une politique d'accès pour accorder Comptes AWS aux utilisateurs, aux groupes et aux rôles IAM l'autorisation d'effectuer des opérations sur une ressource. Par exemple, vous pouvez accorder une `PUT Object` autorisation à un autre compte Compte AWS afin qu'il puisse télécharger des objets dans votre compartiment.

Une politique d'accès décrit qui a accès à quels éléments. Lorsqu'Amazon S3 reçoit une demande, il doit évaluer toutes les politiques d'accès afin de déterminer s'il convient d'autoriser ou de refuser la demande. Pour plus d'informations sur la manière dont Amazon S3 évalue ces politiques, consultez Comment Amazon S3 autorise une demande.

Les outils de gestion des accès disponibles dans Amazon S3 sont les suivants.

Politique de compartiment

Une politique de compartiment Amazon S3 est une politique basée sur les ressources au format JSON AWS Identity and Access Management (IAM) attachée à un compartiment particulier. Utilisez des politiques de compartiment pour accorder des autorisations à d'autres identités Comptes AWS

ou à des identités IAM pour le compartiment et les objets qu'il contient. De nombreux cas d'utilisation de la gestion des accès S3 peuvent être satisfaits à l'aide d'une politique de compartiment. Grâce aux politiques relatives aux compartiments, vous pouvez personnaliser l'accès aux compartiments pour vous assurer que seules les identités que vous avez approuvées peuvent accéder aux ressources et effectuer des actions au sein de celles-ci. Pour plus d'informations, consultez Politiques relatives aux compartiments pour Amazon S3.

Voici un exemple de stratégie de compartiment. Vous exprimez la politique du bucket à l'aide d'un fichier JSON. Cet exemple de politique accorde à un rôle IAM l'autorisation de lecture à tous les objets du compartiment. Il contient une instruction nomméeBucketLevelReadPermissions, qui autorise l's3:GetObjectaction (autorisation de lecture) sur les objets d'un compartiment nomméDOC-EXAMPLE-BUCKET1. En spécifiant un rôle IAM comme étant lePrincipal, cette politique accorde l'accès à tout utilisateur IAM doté de ce rôle. Pour utiliser cet exemple de politique, remplacez *user input placeholders* par vos propres informations.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"BucketLevelReadPermissions",
      "Effect":"Allow",
      "Principal": {
     "AWS": "arn:aws:iam::123456789101:role/s3-role"
      },
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*"]
    }]
  }
```

> **Note**
>
> Lors de la création de politiques, évitez d'utiliser des caractères génériques (*) dans l'élément Principal, car cela permet à quiconque d'accéder effectivement à vos ressources Amazon S3. Répertoriez plutôt explicitement les utilisateurs ou les groupes autorisés à accéder au bucket, ou listez les conditions qui doivent être respectées en utilisant une clause de condition dans la politique. En outre, plutôt que d'inclure un caractère générique pour les actions de vos utilisateurs ou groupes, accordez-leur des autorisations spécifiques le cas échéant.

Politique basée sur l'identité

Une politique utilisateur basée sur l'identité ou IAM est un type de stratégie AWS Identity and Access Management (IAM). Une stratégie basée sur l'identité est une stratégie au format JSON attachée aux utilisateurs, groupes ou rôles IAM de votre compte. AWS Vous pouvez utiliser des politiques basées sur l'identité pour accorder à une identité IAM l'accès à vos compartiments ou à vos objets. Vous pouvez créer des utilisateurs, des groupes et des rôles IAM dans votre compte et leur associer des politiques d'accès. Vous pouvez ensuite accorder l'accès aux AWS ressources, notamment aux ressources Amazon S3. Pour plus d'informations, consultez Politiques basées sur l'identité pour Amazon S3.

Voici un exemple de stratégie basée sur l'identité. L'exemple de politique permet au rôle IAM associé d'effectuer six actions Amazon S3 différentes (autorisations) sur un compartiment et les objets qu'il contient. Si vous associez cette politique à un rôle IAM dans votre compte et que vous attribuez ce rôle à certains de vos utilisateurs IAM, les utilisateurs dotés de ce rôle pourront effectuer ces actions sur les ressources (compartiments) spécifiées dans votre politique. Pour utiliser cet exemple de politique, remplacez *user input placeholders* par vos propres informations.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "AssignARoleActions",
    "Effect": "Allow",
    "Action": [
   "s3:PutObject",
   "s3:GetObject",
   "s3:ListBucket",
   "s3:DeleteObject",
   "s3:GetBucketLocation"
    ],
    "Resource": [
   "arn:aws:s3:::DOC-EXAMPLE-BUCKET1/*",
   "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
    ]
    },
  {
    "Sid": "AssignARoleActions2",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
```

```
  }
  ]
}
```

Octrois d’accès S3

Utilisez S3 Access Grants pour créer des autorisations d'accès à vos données Amazon S3 pour les identités figurant dans les annuaires d'identités d'entrepriseActive Directory, telles que les identités AWS Identity and Access Management (IAM). S3 Access Grants vous aide à gérer les autorisations de données à grande échelle. En outre, S3 Access Grants enregistre l'identité de l'utilisateur final et l'application utilisée pour accéder aux AWS CloudTrail données S3. Cela fournit un historique d'audit détaillé jusqu'à l'identité de l'utilisateur final pour tous les accès aux données de vos compartiments S3. Pour plus d’informations, consultez Gestion de l’accès avec les octrois d’accès S3.

Points d'accès

Amazon S3 Access Points simplifie la gestion de l'accès aux données à grande échelle pour les applications qui utilisent des ensembles de données partagés sur S3. Les points d'accès sont appelés points de terminaison réseau attachés à un bucket. Vous pouvez utiliser les points d'accès pour effectuer des opérations sur des objets S3 à grande échelle, telles que le téléchargement et la récupération d'objets. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de contrôler en détail l'accès à vos objets S3. Les points d'accès S3 peuvent être associés à des buckets dans le même compte ou dans un autre compte fiable. Les politiques de points d'accès sont des politiques basées sur les ressources qui sont évaluées conjointement avec la politique de compartiment sous-jacente. Pour plus d’informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Liste de contrôle d'accès (ACL)

Une ACL est une liste de subventions identifiant le bénéficiaire et l'autorisation accordée. Les ACL accordent des autorisations de lecture ou d'écriture de base à d'autres Comptes AWS personnes. Les listes ACL utilisent un schéma XML spécifique à Amazon S3. Une ACL est un type de politique AWS Identity and Access Management (IAM). Une ACL d'objet est utilisée pour gérer l'accès à un objet, et une ACL de bucket est utilisée pour gérer l'accès à un bucket. Avec les politiques de compartiment, il existe une seule politique pour l'ensemble du compartiment, mais les ACL d'objet sont spécifiées pour chaque objet. Nous vous recommandons de désactiver les ACL, sauf dans des circonstances exceptionnelles où vous devez contrôler l'accès individuellement pour chaque objet.

Pour en savoir plus sur l'utilisation des listes ACL, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Warning**
>
> La majorité des cas d'utilisation modernes d'Amazon S3 ne nécessitent pas l'utilisation d'ACL.

Voici un exemple de liste ACL de compartiment. L'autorisation figurant dans l'ACL indique un propriétaire de compartiment disposant d'une autorisation de contrôle total.

```
<?xml version="1.0" encoding="UTF-8"?>
 <AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
   <ID>Owner-Canonical-User-ID</ID>
   <DisplayName>owner-display-name</DisplayName>
  </Owner>
 <AccessControlList>
  <Grant>
   <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Canonical
 User">
    <ID>Owner-Canonical-User-ID</ID>
    <DisplayName>display-name</DisplayName>
   </Grantee>
   <Permission>FULL_CONTROL</Permission>
  </Grant>
 </AccessControlList>
</AccessControlPolicy>
```

Propriété de l'objet

Pour gérer l'accès à vos objets, vous devez en être le propriétaire. Vous pouvez utiliser le paramètre Object Ownership au niveau du bucket pour contrôler la propriété des objets chargés dans votre bucket. Utilisez également la propriété des objets pour activer les ACL. Par défaut, Object Ownership est défini sur le paramètre imposé par le propriétaire du bucket et toutes les ACL sont désactivées. Lorsque les ACL sont désactivées, le propriétaire du compartiment est propriétaire de tous les objets du compartiment et gère exclusivement l'accès aux données. Pour gérer l'accès, le propriétaire du compartiment utilise des politiques ou un autre outil de gestion des accès, à l'exception des ACL. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

La propriété des objets comporte trois paramètres que vous pouvez utiliser à la fois pour contrôler la propriété des objets chargés dans votre bucket et pour activer les ACL :

ACL désactivées

- Application par le propriétaire du compartiment (par défaut) : les ACL sont désactivées, et le propriétaire du compartiment possède automatiquement tous les objets du compartiment et en a le contrôle total. Les ACL n'affectent pas les autorisations relatives aux données du compartiment S3. Le compartiment utilise des stratégies exclusivement pour définir le contrôle des accès.

ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.
- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

Bonnes pratiques supplémentaires

Envisagez d'utiliser les paramètres et outils de compartiment suivants pour protéger les données en transit et au repos, deux éléments essentiels au maintien de l'intégrité et de l'accessibilité de vos données :

- Bloquer l'accès public — Ne désactivez pas le paramètre par défaut au niveau du compartiment Bloquer l'accès public. Ce paramètre bloque l'accès public à vos données par défaut. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.
- Versionnage S3 : pour garantir l'intégrité des données, vous pouvez implémenter le paramètre de gestion des versions du compartiment S3, qui permet de versionner vos objets au fur et à mesure des mises à jour, au lieu de les remplacer. Vous pouvez utiliser le contrôle de version S3 pour conserver, récupérer et restaurer une version précédente, si nécessaire. Pour en savoir plus sur la gestion des versions S3, veuillez consulter Utilisation de la gestion des versions dans les compartiments S3.
- Verrouillage des objets S3 — Le verrouillage des objets S3 est un autre paramètre que vous pouvez implémenter pour garantir l'intégrité des données. Cette fonctionnalité peut implémenter

un modèle write-once-read-many (WORM) pour stocker des objets de manière immuable. Pour en savoir plus sur le verrouillage d'objet, veuillez consulter Utilisation du verrouillage des objets S3.

- Chiffrement d'objets : Amazon S3 propose plusieurs options de chiffrement d'objets qui protègent les données en transit et au repos. Le chiffrement côté serveur chiffre votre objet avant de l'enregistrer sur les disques de ses centres de données, puis le déchiffre lorsque vous téléchargez les objets. Si vous authentifiez votre demande et que vous disposez des autorisations d'accès, il n'y a aucune différence dans la façon dont vous accédez aux objets chiffrés ou non chiffrés. Pour plus d'informations, consultez Protection des données avec le chiffrement côté serveur. S3 chiffre les objets récemment téléchargés par défaut. Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3. Le chiffrement côté client consiste à chiffrer des données avant de les envoyer à Amazon S3. Pour plus d'informations, consultez Protection des données avec le chiffrement côté client.
- Méthodes de signature — Signature Version 4 est le processus d'ajout d'informations d'authentification aux AWS demandes envoyées par HTTP. Pour des raisons de sécurité, la plupart des demandes AWS doivent être signées avec une clé d'accès, qui consiste en un identifiant de clé d'accès et une clé d'accès secrète. Ces deux clés sont généralement appelées informations d'identification de sécurité. Pour plus d'informations, consultez Authentification des demandes (AWS Signature Version 4) et Processus de signature Signature version 4.

Actions

Pour obtenir la liste complète des autorisations et des clés de condition S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Actions

Les actions AWS Identity and Access Management (IAM) pour Amazon S3 sont les actions possibles qui peuvent être effectuées sur un compartiment ou un objet S3. Vous accordez ces actions aux identités afin qu'elles puissent agir sur vos ressources S3. Des exemples d'actions S3 consistent `s3:GetObject` à lire des objets dans un compartiment et `s3:PutObject` à écrire des objets dans un compartiment.

Clés de condition

Outre les actions, les clés de condition IAM se limitent à accorder l'accès uniquement lorsqu'une condition est remplie. Les clés de condition sont facultatives.

> **Note**
>
> Dans une politique d'accès basée sur les ressources, telle qu'une politique de compartiment, ou dans une stratégie basée sur l'identité, vous pouvez spécifier les éléments suivants :
>
> - Une action ou un ensemble d'actions figurant dans l'`Action`élément de la déclaration de politique.
> - Dans l'`Effect`élément de la déclaration de politique, vous pouvez `Allow` spécifier d'autoriser les actions répertoriées ou de `Deny` bloquer les actions répertoriées. Pour continuer à appliquer la pratique du moindre privilège, les `Deny` énoncés contenus dans l'`Effect`élément de la politique d'accès devraient être aussi larges que possible, et les `Allow` énoncés devraient être aussi restreints que possible. `Deny`les effets associés à l'`s3:*`action constituent un autre bon moyen de mettre en œuvre les meilleures pratiques d'adhésion pour les identités incluses dans les déclarations de conditions de politique.
> - Une clé de condition dans l'`Condition`élément d'une déclaration de politique.

Cas d'utilisation de la gestion des accès

Amazon S3 fournit aux propriétaires de ressources une variété d'outils pour accorder l'accès. L'outil de gestion des accès S3 que vous utilisez dépend des ressources S3 que vous souhaitez partager, des identités auxquelles vous accordez l'accès et des actions que vous souhaitez autoriser ou refuser. Vous souhaiterez peut-être utiliser un ou plusieurs outils de gestion d'accès S3 pour gérer l'accès à vos ressources S3.

Dans la plupart des cas, vous pouvez utiliser une politique d'accès pour gérer les autorisations. Une politique d'accès peut être une politique basée sur les ressources, attachée à une ressource, telle qu'un bucket, ou à une autre ressource Amazon S3 ()Ressources S3. Une politique d'accès peut également être une politique basée sur l'identité, attachée à un utilisateur, un groupe ou un rôle AWS Identity and Access Management (IAM) dans votre compte. Vous constaterez peut-être qu'une politique de compartiment convient mieux à votre cas d'utilisation. Pour plus d’informations, consultez Politiques relatives aux compartiments pour Amazon S3. Par ailleurs, avec AWS Identity and Access Management (IAM), vous pouvez créer des utilisateurs, des groupes et des rôles IAM au sein de votre entreprise et gérer leur accès aux compartiments Compte AWS et aux objets par le biais de politiques basées sur l'identité. Pour plus d’informations, consultez Politiques basées sur l'identité pour Amazon S3.

Pour vous aider à naviguer dans ces options de gestion des accès, vous trouverez ci-dessous des cas d'utilisation courants des clients Amazon S3 et des recommandations pour chacun des outils de gestion des accès S3.

Le Compte AWS propriétaire souhaite partager des buckets uniquement avec les utilisateurs du même compte

Tous les outils de gestion des accès peuvent répondre à ce cas d'utilisation de base. Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Politique de compartiment : si vous souhaitez accorder l'accès à un compartiment ou à un petit nombre de compartiments, ou si vos autorisations d'accès aux compartiments sont similaires d'un compartiment à l'autre, utilisez une politique de compartiment. Avec les politiques de compartiment, vous gérez une politique pour chaque compartiment. Pour plus d’informations, consultez Politiques relatives aux compartiments pour Amazon S3.
- Stratégie basée sur l'identité : si vous avez un très grand nombre de compartiments dotés d'autorisations d'accès différentes pour chaque compartiment et que vous ne devez gérer que quelques rôles d'utilisateur, vous pouvez utiliser une stratégie IAM pour les utilisateurs, les groupes ou les rôles. Les politiques IAM sont également une bonne option si vous gérez l'accès des utilisateurs à d'autres AWS ressources, ainsi qu'aux ressources Amazon S3. Pour plus d’informations, consultez Exemple 1 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations sur un compartiment.
- Attributions d'accès S3 — Vous pouvez utiliser les subventions d'accès S3 pour accorder l'accès à vos compartiments, préfixes ou objets S3. S3 Access Grants vous permet de spécifier différentes autorisations au niveau des objets à grande échelle, tandis que les politiques relatives aux compartiments sont limitées à 20 Ko. Pour plus d’informations, consultez Bien démarrer avec les octrois d’accès S3.
- Points d'accès : vous pouvez utiliser des points d'accès, appelés points de terminaison réseau attachés à un bucket. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de vous donner un contrôle détaillé de l'accès à vos objets S3. Pour plus d’informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Le Compte AWS propriétaire souhaite partager des buckets ou des objets avec des utilisateurs d'un autre AWS compte (comptes croisés)

Pour accorder une autorisation à une autre personne Compte AWS, vous devez utiliser une politique de compartiment ou l'un des outils de gestion des accès recommandés ci-dessous. Vous ne pouvez pas utiliser de politique d'accès basée sur l'identité pour ce cas d'utilisation. Pour plus d'informations sur l'octroi d'un accès entre comptes, consultez Comment fournir un accès entre comptes aux objets qui se trouvent dans des compartiments Amazon S3 ?

Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Politique de compartiment : avec les politiques de compartiment, vous gérez une politique pour chaque compartiment. Pour plus d'informations, consultez Politiques relatives aux compartiments pour Amazon S3.
- Subventions d'accès S3 — Vous pouvez utiliser les subventions d'accès S3 pour accorder des autorisations entre comptes à vos compartiments, préfixes ou objets S3. Vous pouvez utiliser les subventions d'accès S3 pour spécifier différentes autorisations au niveau des objets à grande échelle, tandis que les politiques relatives aux compartiments sont limitées à 20 Ko. Pour plus d'informations, consultez Bien démarrer avec les octrois d'accès S3.
- Points d'accès : vous pouvez utiliser des points d'accès, appelés points de terminaison réseau attachés à un bucket. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de vous donner un contrôle détaillé de l'accès à vos objets S3. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Le Compte AWS propriétaire ou le propriétaire du compartiment doit accorder des autorisations au niveau de l'objet ou du préfixe, et ces autorisations varient d'un objet à l'autre ou d'un préfixe à l'autre

Dans une politique de compartiment, par exemple, vous pouvez autoriser l'accès aux objets d'un compartiment qui partagent un préfixe de nom de clé spécifique ou possèdent une balise spécifique. Vous pouvez accorder une autorisation de lecture aux objets commençant par le préfixe `logs/` du nom de clé. Toutefois, si vos autorisations d'accès varient en fonction de l'objet, il n'est peut-être pas pratique d'accorder des autorisations à des objets individuels à l'aide d'une politique de compartiment, d'autant plus que la taille des politiques de compartiment est limitée à 20 Ko.

Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Attributions d'accès S3 : vous pouvez utiliser les subventions d'accès S3 pour gérer les autorisations au niveau de l'objet ou au niveau du préfixe. Contrairement aux politiques relatives aux compartiments, vous pouvez utiliser S3 Access Grants pour spécifier différentes autorisations au niveau des objets à grande échelle. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Bien démarrer avec les octrois d'accès S3.
- Points d'accès : vous pouvez utiliser des points d'accès pour gérer les autorisations au niveau des objets ou des préfixes. Les points d'accès sont appelés points de terminaison réseau attachés à un bucket. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de vous donner un contrôle détaillé de l'accès à vos objets S3. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.
- ACL — Nous déconseillons d'utiliser des listes de contrôle d'accès (ACL), en particulier parce que les ACL sont limitées à 100 autorisations par objet. Toutefois, si vous choisissez d'activer les ACL, dans les paramètres de votre bucket, définissez Object Ownership sur Bucket owner preferred et sur ACL activées. Avec ce paramètre, de nouveaux objets écrits avec la liste ACL `bucket-owner-full-control` prédéfinie sont automatiquement détenus par le propriétaire du compartiment plutôt que par l'auteur d'objets. Vous pouvez ensuite utiliser les ACL d'objet, qui sont une politique d'accès au format XML, pour autoriser d'autres utilisateurs à accéder à l'objet. Pour plus d'informations, consultez Présentation de la liste de contrôle d'accès (ACL).

Le Compte AWS propriétaire ou le propriétaire du bucket souhaite limiter l'accès au bucket uniquement à des identifiants de compte spécifiques

Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Politique de compartiment : avec les politiques de compartiment, vous gérez une politique pour chaque compartiment. Pour plus d'informations, consultez Politiques relatives aux compartiments pour Amazon S3.
- Points d'accès : les points d'accès sont appelés points de terminaison réseau attachés à un bucket. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de vous donner un contrôle détaillé de l'accès à vos objets S3. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Le Compte AWS propriétaire ou le propriétaire du bucket souhaite disposer de points de terminaison distincts pour chaque utilisateur ou application qui accède à ses données

Nous recommandons l'outil de gestion des accès suivant pour ce cas d'utilisation :

- Points d'accès : les points d'accès sont appelés points de terminaison réseau attachés à un bucket. Un bucket peut contenir jusqu'à 10 000 points d'accès, et pour chaque point d'accès, vous pouvez appliquer des autorisations et des contrôles réseau distincts afin de vous donner un contrôle détaillé de l'accès à vos objets S3. Chaque point d'accès applique une stratégie de point d'accès personnalisée qui fonctionne conjointement avec la stratégie de compartiment associée au compartiment sous-jacent. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Le Compte AWS propriétaire ou le propriétaire du bucket doit gérer l'accès depuis les points de terminaison Virtual Private Cloud (VPC) pour S3

Les points de terminaison Virtual Private Cloud (VPC) pour Amazon S3 sont des entités logiques au sein d'un VPC qui autorisent la connectivité uniquement à S3. Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Compartiments dans un environnement VPC : vous pouvez utiliser une politique de compartiment pour contrôler qui est autorisé à accéder à vos compartiments et à quels points de terminaison VPC ils peuvent accéder. Pour plus d'informations, consultez Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.
- Points d'accès — Si vous choisissez de configurer des points d'accès, vous pouvez utiliser une politique de point d'accès. Vous pouvez configurer n'importe quel point d'accès pour accepter uniquement les demandes provenant d'un cloud privé virtuel (VPC) afin de restreindre l'accès aux données Amazon S3 à un réseau privé. Vous pouvez également configurer des paramètres de blocage de l'accès public personnalisés pour chaque point d'accès. Pour plus d'informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

Le Compte AWS propriétaire ou le propriétaire du bucket doit mettre un site Web statique à la disposition du public

Avec S3, vous pouvez héberger un site Web statique et autoriser tout le monde à consulter le contenu du site Web, qui est hébergé à partir d'un compartiment S3.

Nous recommandons les outils de gestion des accès suivants pour ce cas d'utilisation :

- Amazon CloudFront — Cette solution vous permet d'héberger un site Web statique Amazon S3 accessible au public tout en continuant à bloquer tout accès public au contenu d'un compartiment. Si vous souhaitez conserver les quatre paramètres S3 Block Public Access activés et héberger un site Web statique S3, vous pouvez utiliser le contrôle CloudFront d'accès d'origine (OAC) d'Amazon. Amazon CloudFront fournit les fonctionnalités requises pour configurer un site Web statique sécurisé. En outre, les sites Web statiques Amazon S3 qui n'utilisent pas cette solution ne peuvent prendre en charge que les points de terminaison HTTP. CloudFront utilise le stockage durable d'Amazon S3 tout en fournissant des en-têtes de sécurité supplémentaires, tels que HTTPS. HTTPS accroît la sécurité en chiffrant une demande HTTP normale et en offrant une protection contre les cyberattaques courantes.

  Pour plus d'informations, consultez Getting started with a secure static website in the Amazon CloudFront Developer Guide.

- Rendre votre compartiment Amazon S3 accessible au public : vous pouvez configurer un compartiment pour qu'il soit utilisé comme site Web statique accessible au public.

  > **Warning**
  >
  > Nous ne recommandons pas cette méthode. Nous vous recommandons plutôt d'utiliser les sites Web statiques Amazon S3 dans le cadre d'Amazon CloudFront. Pour plus d'informations, consultez l'option précédente ou la section Commencer à utiliser un site Web statique sécurisé.

  Pour créer un site Web statique Amazon S3, sans Amazon CloudFront, vous devez d'abord désactiver tous les paramètres de blocage de l'accès public. Lorsque vous rédigez la politique de compartiment pour votre site Web statique, veillez à autoriser uniquement des actions `s3:GetObject`, et non pas des autorisations `ListObject` ni `PutObject`. Cela permet de s'assurer que les utilisateurs ne peuvent pas voir tous les objets de votre compartiment ou ajouter leur propre contenu. Pour plus d'informations, consultez Définition des autorisations pour l'accès au site web.

Le Compte AWS propriétaire ou le propriétaire du bucket souhaite rendre le contenu d'un bucket accessible au public

Lors de la création d'un nouveau compartiment Amazon S3, le paramètre Bloquer l'accès public est activé par défaut. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

Nous vous déconseillons d'autoriser l'accès public à votre bucket. Toutefois, si vous devez le faire pour un cas d'utilisation particulier, nous recommandons l'outil de gestion des accès suivant pour ce cas d'utilisation :

- Désactiver le paramètre Bloquer l'accès public : le propriétaire d'un compartiment peut autoriser les demandes non authentifiées adressées au compartiment. Par exemple, les demandes d'objets PUT non authentifiées sont autorisées lorsqu'un bucket dispose d'une politique de bucket public ou lorsqu'une ACL de bucket accorde un accès public. Toutes les demandes non authentifiées sont faites par d'autres AWS utilisateurs arbitraires, ou même par des utilisateurs anonymes non authentifiés. Cet utilisateur est représenté dans les listes ACL par l'ID d'utilisateur canonique spécifique `65a011a29cdf8ec533ec3d1ccaae921c`. Si un objet est chargé vers un `WRITE` ou`FULL_CONTROL`, cela donne spécifiquement accès au groupe Tous les utilisateurs ou à l'utilisateur anonyme. Pour plus d’informations sur les politiques de compartiment public et les listes de contrôle d’accès (ACL) publiques, consultez La signification du mot « public ».

Le Compte AWS propriétaire ou le propriétaire du bucket a dépassé les limites de taille fixées par la politique d'accès

Les politiques de compartiment et les politiques basées sur l'identité ont une limite de taille de 20 Ko. Si vos exigences en matière d'autorisation d'accès sont complexes, vous risquez de dépasser cette limite de taille.

Nous avons recommandé les outils de gestion des accès suivants pour ce cas d'utilisation :

- Points d'accès : utilisez des points d'accès si cela correspond à votre cas d'utilisation. Dans le cas des points d'accès, chaque compartiment possède plusieurs points de terminaison réseau nommés, chacun ayant sa propre politique de point d'accès qui fonctionne avec la politique de compartiment sous-jacente. Toutefois, les points d'accès ne peuvent agir que sur des objets, et non sur des compartiments, et ne prennent pas en charge la réplication entre régions. Pour plus d’informations, consultez Gestion de l'accès aux données avec les points d'accès Amazon S3.

- Subventions d'accès S3 : utilisez les subventions d'accès S3, qui prennent en charge un très grand nombre de subventions donnant accès à des compartiments, à des préfixes ou à des objets. Pour plus d'informations, consultez Bien démarrer avec les octrois d'accès S3.

Le rôle de Compte AWS propriétaire ou d'administrateur souhaite accorder l'accès au bucket, au préfixe ou à l'objet directement aux utilisateurs ou aux groupes d'un annuaire d'entreprise

Au lieu de gérer les utilisateurs, les groupes et les rôles via AWS Identity and Access Management (IAM), vous pouvez y ajouter votre annuaire d' AWS IAM Identity Center entreprise. Pour plus d'informations, consultez Qu'est-ce qu'IAM Identity Center ? .

Après avoir ajouté votre annuaire d'entreprise à AWS IAM Identity Center, nous vous recommandons d'utiliser l'outil de gestion des accès suivant pour accorder aux identités d'annuaire d'entreprise l'accès à vos ressources S3 :

- Subventions d'accès S3 : utilisez les subventions d'accès S3, qui permettent d'accorder l'accès à des utilisateurs ou à des rôles dans votre annuaire d'entreprise. Pour plus d'informations, consultez Bien démarrer avec les octrois d'accès S3.

Le Compte AWS propriétaire ou le propriétaire du compartiment souhaite autoriser le AWS CloudFront service à écrire CloudFront des journaux dans un compartiment S3

Nous avons recommandé l'outil de gestion des accès suivant pour ce cas d'utilisation :

- ACL de compartiment — Le seul cas d'utilisation recommandé pour les ACL de compartiment est d'accorder des autorisations à certaines Services AWS, comme le CloudFront `awslogsdelivery` compte Amazon. Lorsque vous créez ou mettez à jour une distribution et que vous activez la CloudFront journalisation, CloudFront met à jour l'ACL du bucket pour `awslogsdelivery` autoriser `FULL_CONTROL` le compte à écrire des journaux dans votre bucket. Pour plus d'informations, consultez la section Autorisations requises pour configurer la journalisation standard et pour accéder à vos fichiers journaux dans le manuel Amazon CloudFront Developer Guide.
  Si le compartiment qui stocke les journaux utilise le paramètre imposé par le propriétaire du compartiment pour S3 Object Ownership afin de désactiver les ACL, il CloudFront ne peut pas écrire de journaux dans le compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

En tant que propriétaire du compartiment, vous souhaitez conserver le contrôle total des objets ajoutés au compartiment par d'autres utilisateurs

Vous pouvez autoriser d'autres comptes à télécharger des objets dans votre compartiment en utilisant une politique de compartiment, un point d'accès ou des autorisations d'accès S3. Si vous avez accordé un accès multicompte à votre compartiment, vous pouvez vous assurer que tous les objets chargés dans votre compartiment restent sous votre contrôle total.

Nous avons recommandé l'outil de gestion des accès suivant pour ce cas d'utilisation :

- Propriété de l'objet : maintenez le paramètre Propriété de l'objet au niveau du compartiment au niveau du paramètre par défaut imposé par le propriétaire du compartiment.

Résolution des problèmes de gestion des accès

Les ressources suivantes peuvent vous aider à résoudre les problèmes liés à la gestion des accès S3 :

Résolution des erreurs d'accès refusé (403 – Interdit)

Si vous rencontrez des problèmes de refus d'accès, vérifiez les paramètres au niveau du compte et au niveau du compartiment. Vérifiez également la fonctionnalité de gestion des accès que vous utilisez pour accorder l'accès afin de vous assurer que la politique, le paramètre ou la configuration sont corrects. Pour plus d'informations sur les causes courantes des erreurs d'accès refusé (403 – Interdit) dans Amazon S3, consultez Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3.

Analyseur d'accès IAM pour S3

Si vous ne souhaitez rendre aucune de vos ressources accessible au public, ou si vous souhaitez limiter l'accès public à vos ressources, vous pouvez utiliser IAM Access Analyzer pour S3. Sur la console Amazon S3, utilisez IAM Access Analyzer for S3 pour examiner tous les compartiments dotés de listes de contrôle d'accès aux compartiments (ACL), de politiques de compartiment ou de politiques de point d'accès qui accordent un accès public ou partagé. IAM Access Analyzer for S3 vous avertit de la présence de compartiments configurés pour autoriser l'accès à toute personne sur Internet ou autre Comptes AWS, y compris Comptes AWS en dehors de votre organisation. Pour chaque compartiment public ou partagé, vous recevez des résultats qui signalent la source et le niveau d'accès public ou partagé.

Dans IAM Access Analyzer pour S3, vous pouvez bloquer tout accès public à un bucket en une seule action. Nous vous recommandons de bloquer tout accès public à vos buckets, sauf si vous avez besoin d'un accès public pour prendre en charge un cas d'utilisation spécifique. Avant de bloquer tout accès public, assurez-vous que vos applications continueront de fonctionner correctement sans accès public. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Vous pouvez également consulter vos paramètres d'autorisation au niveau du compartiment pour configurer des niveaux d'accès détaillés. Pour les cas d'utilisation spécifiques et vérifiés nécessitant un accès public ou partagé, vous pouvez confirmer et enregistrer votre intention de maintenir le niveau d'accès public ou partagé en archivant les résultats pour le compartiment. Vous pouvez revisiter et modifier ces configurations de compartiments à tout moment. Vous pouvez également télécharger vos résultats sous forme de rapport CSV à des fins d'audit.

L'analyseur d'accès IAM pour S3 est disponible gratuitement sur la console Amazon S3. L'analyseur d'accès IAM pour S3 est optimisé par l'analyseur d'accès IAM d' AWS Identity and Access Management (IAM). Pour utiliser IAM Access Analyzer for S3 sur la console Amazon S3, vous devez vous rendre sur la console IAM et créer un analyseur au niveau du compte dans IAM Access Analyzer pour chaque région individuelle.

Pour plus d'informations sur l'analyseur d'accès IAM pour S3, consultez Examen de l'accès aux compartiments à l'aide de l'analyseur d'accès IAM pour S3.

Journalisation et surveillance

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de vos solutions Amazon S3 afin que vous puissiez corriger plus facilement une défaillance d'accès. La journalisation peut fournir un aperçu des erreurs reçues par les utilisateurs, ainsi que du moment et des demandes qui leur sont adressées. AWS fournit plusieurs outils pour surveiller vos ressources Amazon S3, tels que les suivants :

- AWS CloudTrail
- Journaux d'accès Amazon S3
- AWS Trusted Advisor
- Amazon CloudWatch

Pour plus d'informations, consultez Journalisation et surveillance dans Amazon S3.

Identity and Access Management pour Amazon S3

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources Amazon S3. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans Amazon S3.

Utilisateur du service : si vous utilisez le service Amazon S3 pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin.

Au fur et à mesure que vous utilisez de plus en plus de fonctionnalités d'Amazon S3 pour effectuer votre travail, il se peut que vous ayez besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité d'Amazon S3, consultezRésolution des problèmes d'identité et d'accès à Amazon S3.

Administrateur du service — Si vous êtes responsable des ressources Amazon S3 au sein de votre entreprise, vous avez probablement un accès complet à Amazon S3. C'est à vous de déterminer les fonctionnalités et les ressources d'Amazon S3 auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec Amazon S3, consultezComment Amazon S3 fonctionne avec IAM.

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à Amazon S3. Pour consulter des exemples de politiques basées sur l'identité Amazon S3 que vous pouvez utiliser dans IAM, consultez. Politiques basées sur l'identité pour Amazon S3

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide

de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour

obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour

obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
  - Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés

à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d’accès au sein de l’instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d’instance contient le rôle et permet aux programmes qui s’exécutent sur l’instance EC2 d’obtenir des informations d’identification temporaires. Pour plus d’informations, consultez Utilisation d’un rôle IAM pour accorder des autorisations à des applications s’exécutant sur des instances Amazon EC2 dans le Guide de l’utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d’un utilisateur) dans le Guide de l’utilisateur IAM.

Gestion des accès à l’aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d’informations sur la structure et le contenu des documents de politique JSON, consultez Vue d’ensemble des politiques JSON dans le Guide de l’utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C’est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d’aucune autorisation. Pour octroyer aux utilisateurs des autorisations d’effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L’administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d’une action, quelle que soit la méthode que vous utilisez pour exécuter l’opération. Par exemple, supposons que vous disposiez d’une politique qui

autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux

politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez politiques de séance dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s’appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment Amazon S3 fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Amazon S3, découvrez quelles fonctionnalités IAM peuvent être utilisées avec Amazon S3.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon S3

Fonction IAM	Prise en charge d'Amazon S3
Politiques basées sur l’identité	Oui
Politiques basées sur les ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACL	Oui
ABAC (identifications dans les politiques)	Partielle
Informations d’identification temporaires	Oui
Transmission des sessions d’accès (FAS)	Oui
Fonctions de service	Oui
Rôles liés à un service	Partielle

Pour obtenir une vue d'ensemble de la façon dont Amazon S3 et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez les AWS services compatibles avec IAM dans le guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon S3

Prend en charge les politiques basées sur l’identité	Oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Création de politiques IAM dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Exemples de politiques basées sur l'identité pour Amazon S3

Pour consulter des exemples de politiques basées sur l'identité Amazon S3, consultez. Politiques basées sur l'identité pour Amazon S3

Politiques basées sur les ressources au sein d'Amazon S3

Prend en charge les politiques basées sur les ressources	Oui

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour

contrôler l’accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d’actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L’ajout d’un principal entre comptes à une politique basée sur les ressources ne représente qu’une partie de l’instauration de la relation d’approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l’entité. Toutefois, si une politique basée sur des ressources accorde l’accès à un principal dans le même compte, aucune autre politique basée sur l’identité n’est requise. Pour plus d'informations, consultez la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM.

Le service Amazon S3 prend en charge les politiques relatives aux compartiments, les politiques relatives aux points d'accès et les autorisations d'accès :

- Les politiques de compartiment sont des politiques basées sur les ressources associées à un compartiment Amazon S3. Une politique de compartiment définit les principaux autorisés à effectuer des actions sur le compartiment.
- Les politiques de point d'accès sont des politiques basées sur les ressources qui sont évaluées conjointement avec la politique de compartiment sous-jacente.
- Les autorisations d'accès constituent un modèle simplifié permettant de définir les autorisations d'accès aux données dans Amazon S3 par préfixe, compartiment ou objet. Pour plus d'informations sur les subventions d'accès S3, consultezGestion de l’accès avec les octrois d’accès S3.

Principes relatifs aux politiques relatives aux compartiments

L'`Principal`élément indique l'utilisateur, le compte, le service ou toute autre entité à laquelle l'accès à une ressource est autorisé ou refusé. Voici quelques exemples de spécification de `Principal`. Pour plus d’informations, consultez Principal dans le Guide de l’utilisateur IAM.

Accorder des autorisations à un Compte AWS

Pour accorder des autorisations à un Compte AWS, identifiez le compte en utilisant le format suivant.

```
"AWS":"account-ARN"
```

Voici quelques exemples.

```
"Principal":{"AWS":"arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal":{"AWS":
["arn:aws:iam::AccountID1WithoutHyphens:root","arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

Octroyer des autorisations à un utilisateur IAM

Pour accorder une autorisation à un utilisateur IAM de votre compte, vous devez fournir une paire de nom-valeur `"AWS":"user-ARN"`.

```
"Principal":{"AWS":"arn:aws:iam::account-number-without-hyphens:user/username"}
```

Pour des exemples détaillés fournissant des step-by-step instructions, reportez-vous aux sections Exemple 1 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations sur un compartiment etExemple 3 : propriétaire d'un compartiment accordant des autorisations sur des objets qu'il ne possède pas.

> **Note**
>
> Si une identité IAM est supprimée après la mise à jour de votre politique de compartiment, la politique de compartiment affichera un identifiant unique dans l'élément principal au lieu d'un ARN. Ces identifiants uniques ne sont jamais réutilisés. Vous pouvez donc supprimer en toute sécurité les principaux dotés d'identifiants uniques de toutes vos déclarations de police. Pour plus d'informations sur les identifiants uniques, consultez Identificateurs IAM dans le Guide de l'utilisateur IAM.

Octroyer des autorisations anonymes

> **Warning**
>
> Soyez vigilant lorsque vous accordez un accès anonyme à votre compartiment Amazon S3. Lorsque vous accordez un accès anonyme, tout le monde peut accéder à votre

compartiment. Nous vous recommandons vivement de ne jamais accorder un type d'accès en écriture anonyme quel qu'il soit à votre compartiment S3.

Pour accorder l'autorisation à tout le monde (on parle aussi d'accès anonyme), définissez le caractère générique `"*"` en tant que valeur `Principal`. Par exemple, si vous configurez votre compartiment en tant que site Web, vous voulez que tous les objets dans le compartiment soit accessibles publiquement.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

L'utilisation `"Principal": "*"` avec `Allow` effet dans le cadre d'une politique basée sur les ressources permet à quiconque, même s'il n'est pas connecté AWS, d'accéder à votre ressource.

L'utilisation de l'interface `"Principal" : { "AWS" : "*" }` avec un effet `Allow` dans une politique basée sur les ressources permet à n'importe quel utilisateur racine, utilisateur IAM, séance à rôle supposé ou utilisateur fédéré dans n'importe quel compte de la même partition d'accéder à votre ressource.

Pour les utilisateurs anonymes, ces deux méthodes sont équivalentes. Pour plus d'informations, consultez Tous les principaux dans le Guide de l'utilisateur IAM.

Vous ne pouvez pas utiliser un caractère générique pour une correspondance à une partie d'un nom ou d'un ARN principal.

> **Important**
>
> Comme tout le monde peut créer un Compte AWS, le niveau de sécurité de ces deux méthodes est équivalent, même si elles fonctionnent différemment.

Restriction des autorisations sur les ressources

Vous pouvez également utiliser la politique sur les ressources pour restreindre l'accès aux ressources qui seraient autrement accessibles aux principaux IAM. Utilisez une instruction `Deny` pour empêcher l'accès.

L'exemple suivant bloque l'accès si aucun protocole de transport sécurisé n'est utilisé :

```
{"Effect": "Deny",
 "Principal": "*",
 "Action": "s3:*",
 "Resource": <bucket ARN>,
 "Condition": {
   "Boolean": { "aws:SecureTransport" : "false"}
 }
}
```

Au lieu de tenter de refuser l'accès à des comptes ou des principaux spécifiques selon cette méthode, l'utilisation de `"Principal": "*"` de sorte que cette restriction s'applique à tout le monde est une bonne pratique pour cette politique.

Exiger un accès via des CloudFront URL

Vous pouvez demander à vos utilisateurs d'accéder à votre contenu Amazon S3 uniquement en utilisant des CloudFront URL plutôt que des URL Amazon S3. Pour ce faire, créez un contrôle CloudFront d'accès à l'origine (OAC). Modifiez ensuite les autorisations sur vos données S3. Dans votre politique de compartiment, vous pouvez définir le CloudFront rôle principal comme suit :

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Utilisez un `Condition` élément de la politique CloudFront pour autoriser l'accès au compartiment uniquement lorsque la demande provient de la CloudFront distribution contenant l'origine S3.

```
       "Condition": {
          "StringEquals": {
             "AWS:SourceArn":
 "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
          }
       }
```

Pour plus d'informations sur l'obligation d'accéder à S3 via CloudFront des URL, consultez Restreindre l'accès à une origine Amazon Simple Storage Service dans le manuel Amazon CloudFront Developer Guide. Pour plus d'informations sur les avantages en matière de sécurité et de confidentialité liés à l'utilisation d'Amazon CloudFront, consultez Configuration de l'accès sécurisé et restriction de l'accès au contenu.

Exemples de politiques basées sur les ressources pour Amazon S3

- Pour consulter des exemples de politiques relatives aux compartiments Amazon S3, consultezPolitiques relatives aux compartiments pour Amazon S3.
- Pour consulter des exemples de politiques relatives aux points d'accès, voirConfiguration des stratégies IAM pour l'utilisation des points d'accès.

Actions politiques pour Amazon S3

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Ce qui suit montre les différents types de relations de mappage entre les opérations de l'API S3 et les actions politiques requises.

- O ne-to-one mappage du même nom. Par exemple, pour utiliser l'opération `PutBucketPolicy` API, l'action `s3:PutBucketPolicy` politique est requise.
- O ne-to-one mappage avec des noms différents. Par exemple, pour utiliser l'opération `ListObjectsV2` API, l'action `s3:ListBucket` politique est requise.
- Sur la ne-to-many cartographie. Par exemple, pour utiliser l'opération `HeadObject` API, le `s3:GetObject` est requis. En outre, lorsque vous utilisez S3 Object Lock et que vous souhaitez obtenir le statut de conservation légale ou les paramètres de rétention d'un objet, les actions de `s3:GetObjectRetention` politique `s3:GetObjectLegalHold` ou de politique correspondantes sont également requises avant de pouvoir utiliser l'opération d'`HeadObject`API.

- any-to-one Cartographie M. Par exemple, pour utiliser les opérations `ListObjectsV2` d'`HeadBucket`API ou, l'action `s3:ListBucket` de politique est requise.

Pour consulter la liste des actions Amazon S3 à utiliser dans les politiques, consultez la section Actions définies par Amazon S3 dans le Service Authorization Reference. Pour obtenir la liste complète des opérations d'API Amazon S3, consultez la section Actions d'API Amazon S3 dans le manuel Amazon Simple Storage Service API Reference.

Les actions politiques dans Amazon S3 utilisent le préfixe suivant avant l'action :

```
s3
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "s3:action1",
      "s3:action2"
         ]
```

Opérations de compartiment

Les opérations de compartiment sont des opérations d'API S3 qui opèrent sur le type de ressource de compartiment. Par exemple, `CreateBucket`, `ListObjectsV2` et `PutBucketPolicy`. Les actions de politique S3 pour les opérations de compartiment nécessitent que l'`Resource`élément des politiques de compartiment ou des politiques basées sur l'identité IAM soit l'identifiant Amazon Resource Name (ARN) du type de compartiment S3 dans l'exemple de format suivant.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
```

La politique de compartiment suivante accorde à l'utilisateur *Akua possédant* le compte *12345678901* l'`s3:ListBucket`autorisation d'effectuer l'opération d'API ListObjectsV2 et de répertorier les objets dans un compartiment S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "Allow Akua to list objects in the bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        }
    ]
}
```

Opérations relatives aux compartiments dans les politiques de point d'accès

Les autorisations accordées dans le cadre d'une politique de point d'accès ne sont effectives que si le compartiment sous-jacent autorise les mêmes autorisations. Lorsque vous utilisez des points d'accès S3, vous devez déléguer le contrôle d'accès du compartiment au point d'accès ou ajouter les mêmes autorisations dans les politiques du point d'accès à la politique du compartiment sous-jacent. Pour plus d'informations, consultez Configuration des stratégies IAM pour l'utilisation des points d'accès. Dans les politiques de point d'accès, les actions de politique S3 pour les opérations de compartiment nécessitent que vous utilisiez l'`accesspoint`ARN de l'`Resource`élément au format suivant.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La politique de point d'accès suivante accorde à l'utilisateur *Akua* possédant le compte *12345678901* l'`s3:ListBucket`autorisation d'effectuer l'opération d'API ListObjectsV2 via le point d'accès S3 *DOC-EXAMPLE-ACCESS-POINT afin de répertorier les objets dans le compartiment associé au point* d'accès.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
```

```
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-
ACCESS-POINT"
        }
    ]
}
```

> **Note**
>
> Toutes les opérations de compartiment ne sont pas prises en charge par S3 Access Point. Pour plus d'informations, consultez Compatibilité des points d'accès avec les opérations S3.

Opérations sur les objets

Les opérations d'objet sont des opérations d'API S3 qui agissent sur le type de ressource de l'objet. Par exemple, `GetObject`, `PutObject` et `DeleteObject`. Les actions de politique S3 pour les opérations sur les objets nécessitent que l'`Resource`élément des politiques soit l'ARN de l'objet S3 dans les exemples de formats suivants.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/prefix/*"
```

> **Note**
>
> L'ARN de l'objet doit contenir une barre oblique après le nom du bucket, comme indiqué dans les exemples précédents.

La politique de compartiment suivante accorde à l'utilisateur *Akua possédant* le compte *12345678901* l'`s3:PutObject`autorisation d'effectuer l'opération d'PutObjectAPI pour télécharger des objets dans un compartiment S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to upload objects",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Opérations sur les objets dans les politiques de point d'accès

Lorsque vous utilisez des points d'accès S3 pour contrôler l'accès aux opérations sur les objets, vous pouvez utiliser des politiques de point d'accès. Lorsque vous utilisez des politiques de point d'accès, les actions de politique S3 pour les opérations sur les objets nécessitent que vous utilisiez l'`accesspoint`ARN de l'`Resource`élément au format suivant :`arn:aws:s3:region:account-id:accesspoint/access-point-name/object/resource`. Pour les opérations d'objet qui utilisent un point d'accès, vous devez inclure la `/object/` valeur après l'ARN complet du point d'accès dans l'`Resource`élément. Voici quelques exemples.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT/
object/*"
```

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT/
object/prefix/*"
```

La politique de point d'accès suivante accorde à l'utilisateur *Akua possédant* le compte *12345678901* l'`s3:GetObject`autorisation d'effectuer l'opération d'GetObjectAPI via le point d'accès *DOC-EXAMPLE-ACCESS-POINT sur tous les objets du bucket associé au point* d'accès.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to get objects through access point",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-
ACCESS-POINT/object/*"
        }
    ]
}
```

> **Note**
>
> Toutes les opérations sur les objets ne sont pas prises en charge par S3 Access Point. Pour plus d'informations, consultez Compatibilité des points d'accès avec les opérations S3.

Opérations des points d'accès

Les opérations de point d'accès sont des opérations d'API S3 qui opèrent sur le type de `accesspoint` ressource. Par exemple, `CreateAccessPoint`, `DeleteAccessPoint` et `GetAccessPointPolicy`. Les actions de politique S3 pour les opérations de point d'accès ne peuvent être utilisées que dans les politiques basées sur l'identité IAM, et non dans les politiques de compartiment ou les politiques de point d'accès. Les opérations sur les points d'accès nécessitent que l'`Resource`élément soit l'`accesspoint`ARN dans l'exemple de format suivant.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/DOC-EXAMPLE-ACCESS-POINT"
```

La politique basée sur l'identité IAM suivante accorde l'`s3:GetAccessPointPolicy`autorisation d'effectuer l'opération d'`GetAccessPointPolicy`API sur le point d'accès S3 DOC-EXAMPLE-ACCESS-POINT.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant permission to retrieve the access point policy of access
 point DOC-EXAMPLE-ACCESS-POINT",
            "Effect": "Allow",
            "Action": [
```

```
            "s3:GetAccessPointPolicy"
            ],
            "Resource": "arn:aws:s3:*:123456789012:access point/DOC-EXAMPLE-ACCESS-
POINT"
        }
    ]
}
```

Lorsque vous utilisez des points d'accès, pour contrôler l'accès aux opérations du bucket, voir Opérations relatives aux compartiments dans les politiques de point d'accès ; pour contrôler l'accès aux opérations sur les objets, voirOpérations sur les objets dans les politiques de point d'accès. Pour plus d'informations sur la configuration des politiques de point d'accès, consultezConfiguration des stratégies IAM pour l'utilisation des points d'accès.

Opérations du point d'accès Object Lambda

Amazon S3 Object Lambda vous donne la possibilité d'ajouter votre propre code aux requêtes Amazon S3 `GET`, `LIST` et `HEAD` afin de modifier et de traiter les données lorsqu'elles sont renvoyées vers une application. Vous pouvez effectuer des demandes via un point d'accès Object Lambda, qui fonctionne de la même manière que les demandes via d'autres points d'accès. Pour plus d'informations, consultez Transformation d'objets avec S3 Object Lambda.

Pour plus d'informations sur la façon de configurer les politiques pour les opérations du point d'accès Object Lambda, consultez. Configuration des politiques IAM pour les points d'accès Object Lambda

Opérations des points d'accès multirégionaux

Un point d'accès multirégional fournit un point de terminaison global que les applications peuvent utiliser pour répondre aux demandes provenant de compartiments S3 situés dans plusieurs compartiments. Région AWS Vous pouvez utiliser un point d'accès multirégional pour créer des applications multirégionales avec la même architecture que celle utilisée dans une seule région, puis exécuter ces applications n'importe où dans le monde. Pour plus d'informations, consultez Points d'accès multi-régions dans Amazon S3.

Pour plus d'informations sur la façon de configurer des politiques pour les opérations de points d'accès multirégionaux, consultezExemples de politique de point d'accès multi-régions.

Opérations de tâches par lots

Les opérations de travail (Batch Operations) sont des opérations d'API S3 qui opèrent sur le type de ressource de travail. Par exemple : `DescribeJob` et `CreateJob`. Les actions de politique S3

pour les opérations de travail ne peuvent être utilisées que dans les politiques basées sur l'identité IAM, et non dans les politiques de compartiment. En outre, les opérations de travail nécessitent que l'`Resource`élément des politiques basées sur l'identité IAM soit l'`job`ARN dans l'exemple de format suivant.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

La politique basée sur l'identité IAM suivante accorde l's3:DescribeJobautorisation d'effectuer l'opération d'DescribeJobAPI sur S3 Batch Operations Job DOC-EXAMPLE-JOB.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow describing the Batch operation job DOC-EXAMPLE-JOB",
            "Effect": "Allow",
            "Action": [
            "s3:DescribeJob"
            ],
            "Resource": "arn:aws:s3:*:123456789012:job/DOC-EXAMPLE-JOB"
        }
    ]
}
```

Opérations de configuration de S3 Storage Lens

Pour plus d'informations sur la configuration des opérations de configuration de S3 Storage Lens, consultezAutorisations Amazon S3 Storage Lens.

Opérations du compte

Les opérations de compte sont des opérations d'API S3 qui opèrent au niveau du compte. Par exemple, `GetPublicAccessBlock` (pour le compte). Le compte n'est pas un type de ressource défini par Amazon S3. Les actions de politique S3 pour les opérations de compte ne peuvent être utilisées que dans les politiques basées sur l'identité IAM, et non dans les politiques de compartiment. En outre, les opérations de compte nécessitent que l'`Resource`élément des politiques basées sur l'identité IAM soit. `"*"`

La politique basée sur l'identité IAM suivante accorde l'`s3:GetAccountPublicAccessBlock`autorisation d'effectuer l'opération

d'GetPublicAccessBlockAPI au niveau du compte et de récupérer les paramètres de blocage de l'accès public au niveau du compte.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Allow retrieving the account-level Public Access Block settings",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Exemples de politiques pour Amazon S3

- Pour consulter des exemples de politiques basées sur l'identité Amazon S3, consultez. Politiques basées sur l'identité pour Amazon S3
- Pour consulter des exemples de politiques basées sur les ressources Amazon S3, consultez Politiques relatives aux compartiments pour Amazon S3 et. Configuration des stratégies IAM pour l'utilisation des points d'accès

Ressources relatives aux politiques pour Amazon S3

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions

qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Certaines actions de l'API Amazon S3 prennent en charge plusieurs ressources. Par exemple, `s3:GetObject` accède à EXAMPLE-RESOURCE-1 et EXAMPLE-RESOURCE-2, de sorte qu'un principal doit être autorisé à accéder aux deux ressources. Pour spécifier plusieurs ressources dans une seule instruction, séparez leurs ARN par des virgules.

```
"Resource": [
      "EXAMPLE-RESOURCE-1",
      "EXAMPLE-RESOURCE-2"
```

Les ressources d'Amazon S3 sont des compartiments, des objets, des points d'accès ou des tâches. Dans une politique, utilisez l'Amazon Resource Name (ARN) du compartiment, de l'objet, du point d'accès ou de la tâche pour identifier la ressource.

Pour consulter la liste complète des types de ressources Amazon S3 et de leurs ARN, consultez la section Ressources définies par Amazon S3 dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon S3.

Wildcards pour les ARN des ressources

Il est possible d'utiliser des caractères génériques dans l'ARN d'une ressource. Vous pouvez utiliser des caractères génériques (* et ?) dans n'importe quel segment d'ARN (les parties séparées par des signes deux-points). Un astérisque (*) représente n'importe quelle combinaison de zéro ou de plusieurs caractères, et un point d'interrogation (?) représente un seul caractère quelconque. Vous pouvez utiliser plusieurs caractères * ou ? dans segment, mais un caractère générique ne peut pas s'étendre à plusieurs segments.

- L'ARN suivant utilise le caractère générique * dans la partie relative-ID de l'ARN pour identifier tous les objets du compartiment `examplebucket`.

```
arn:aws:s3:::examplebucket/*
```

- L'ARN suivant est utilisé * pour indiquer tous les compartiments et objets S3.

```
arn:aws:s3:::*
```

- L'ARN suivant utilise les deux caractères génériques, * et ?, dans la partie `relative-ID`. Il identifie tous les objets des compartiments, par exemple `example1bucket`, `example2bucket`, `example3bucket`, etc.

```
arn:aws:s3:::example?bucket/*
```

Variables de politique pour les ARN des ressources

Vous pouvez utiliser des variables de stratégie dans les ARN Amazon S3. Lors de l'évaluation de la stratégie, ces variables prédéfinies sont remplacées par leurs valeurs correspondantes. Supposons que vous organisez votre compartiment comme un ensemble de dossiers, avec un dossier pour chacun de vos utilisateurs. Le nom de dossier est le même que le nom utilisateur. Pour octroyer aux utilisateurs des autorisations à leurs dossiers, vous pouvez spécifier une variable de stratégie dans l'ARN de la ressource :

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

Lors de l'exécution, lorsque la politique est évaluée, la variable `${aws:username}` de l'ARN de la ressource est remplacée par le nom d'utilisateur de la personne qui fait la demande.

Exemples de politiques pour Amazon S3

- Pour consulter des exemples de politiques basées sur l'identité Amazon S3, consultez. Politiques basées sur l'identité pour Amazon S3
- Pour consulter des exemples de politiques basées sur les ressources Amazon S3, consultez Politiques relatives aux compartiments pour Amazon S3 et. Configuration des stratégies IAM pour l'utilisation des points d'accès

Clés de conditions de politique pour Amazon S3

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Chaque clé de condition Amazon S3 correspond au même nom que l'en-tête de demande autorisé par l'API sur laquelle la condition peut être définie. Les clés de condition spécifiques à Amazon S3 dictent le comportement des en-têtes de demande du même nom. Par exemple, la clé de condition `s3:VersionId` utilisée pour accorder une autorisation conditionnelle définit le `s3:GetObjectVersion`
comportement du paramètre de `versionId` requête que vous définissez dans une demande GET Object.

Pour consulter la liste des clés de condition Amazon S3, consultez la section Clés de condition pour Amazon S3 dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon S3.

Exemple : restriction des téléchargements d'objets vers des objets dotés d'une classe de stockage spécifique

Supposons que le compte A, représenté par l'ID de compte 123456789012, possède un compartiment. L'administrateur du compte A souhaite restreindre Dave, un utilisateur du compte A, afin que Dave ne puisse télécharger des objets que dans le compartiment stocké avec la classe `STANDARD_IA` de stockage. Pour restreindre les chargements d'objets vers une classe de stockage spécifique, l'administrateur de compte A peut utiliser la clé de condition `s3:x-amz-storage-class`, comme illustré dans l'exemple de stratégie de compartiment suivant.

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "statement1",
                    "Effect": "Allow",
                    "Principal": {
                      "AWS": "arn:aws:iam::123456789012:user/Dave"
                    },
                    "Action": "s3:PutObject",
                    "Resource": "arn:aws:s3:::example-s3-bucket1/*",
                    "Condition": {
                      "StringEquals": {
                        "s3:x-amz-storage-class": [
                          "STANDARD_IA"
                        ]
                      }
                    }
                  }
                ]
            }
```

Dans l'exemple, le bloc `Condition` spécifie la condition `StringEquals` qui est appliquée à la paire clé-valeur spécifiée, `"s3:x-amz-acl":["public-read"]`. Un jeu de clés prédéfinis peut être utilisé pour exprimer une condition. L'exemple utilise la clé de condition `s3:x-amz-acl`. Cette condition exige que l'utilisateur inclue l'en-tête `x-amz-acl` avec la valeur `public-read` dans chaque demande PUT object.

Exemples de politiques pour Amazon S3

- Pour consulter des exemples de politiques basées sur l'identité Amazon S3, consultez. Politiques basées sur l'identité pour Amazon S3
- Pour consulter des exemples de politiques basées sur les ressources Amazon S3, consultez Politiques relatives aux compartiments pour Amazon S3 et. Configuration des stratégies IAM pour l'utilisation des points d'accès

ACL dans Amazon S3

Prend en charge les listes ACL	Oui

Dans Amazon S3, les listes de contrôle d'accès (ACL) contrôlent les Comptes AWS personnes autorisées à accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

> **Important**
>
> La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL.

Pour plus d'informations sur l'utilisation des ACL pour contrôler l'accès dans Amazon S3, consultezGestion des accès à l'aide des listes ACL.

ABAC avec Amazon S3

Prise en charge d’ABAC (identifications dans les politiques)	Partielle

Le contrôle d’accès basé sur les attributs (ABAC) est une politique d’autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L’étiquetage des entités et des ressources est la première étape d’ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l’identification du principal correspond à celle de la ressource à laquelle il tente d’accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Pour consulter des exemples de politiques basées sur l'identité permettant de limiter l'accès aux tâches S3 Batch Operations en fonction de balises, consultez. Contrôle des autorisations pour les opérations par lot S3 à l'aide d'étiquettes de tâche

ABAC et balises d'objets

Dans les politiques ABAC, les objets utilisent des `s3:` balises plutôt que des `aws:` balises. Pour contrôler l'accès aux objets en fonction des balises d'objets, vous devez fournir les informations relatives aux balises dans l'élément de condition d'une politique à l'aide des balises suivantes :

- `s3:ExistingObjectTag/tag-key`
- `s3:s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/tag-key`

Pour plus d'informations sur l'utilisation de balises d'objet pour contrôler l'accès, y compris des exemples de politiques d'autorisation, consultezStratégies de balisage et de contrôle d'accès.

Utilisation d'informations d'identification temporaires avec Amazon S3

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Transférer les sessions d'accès pour Amazon S3

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

- Le FAS est utilisé par Amazon S3 pour effectuer des appels AWS KMS afin de déchiffrer un objet lorsque SSE-KMS a été utilisé pour le chiffrer. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

- S3 Access Grants utilise également le FAS. Après avoir créé une autorisation d'accès à vos données S3 pour une identité particulière, le bénéficiaire demande un identifiant temporaire à S3 Access Grants. S3 Access Grants obtient un identifiant temporaire pour le demandeur AWS STS et le transmet au demandeur. Pour plus d'informations, consultez Demande d'un accès aux données Amazon S3 via les octrois d'accès S3.

Rôles de service pour Amazon S3

Prend en charge les fonctions du service	Oui

Une fonction de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités d'Amazon S3. Modifiez les rôles de service uniquement lorsque Amazon S3 fournit des instructions à cet effet.

Rôles liés à un service pour Amazon S3

Prend en charge les rôles liés à un service	Partielle

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Amazon S3 prend en charge les rôles liés au service pour Amazon S3 Storage Lens. Pour en savoir plus sur la création ou la gestion des rôles liés aux services Amazon S3, consultez. Utilisation des rôles liés à un service pour le cadre de stockage Amazon S3

Amazon S3 Service en tant que principal

Nom du service dans la politique	Fonctionnalité S3	En savoir plus
`s3.amazonaws.com`	Réplication S3	Configuration de la réplication en direct
`s3.amazonaws.com`	Notifications d'événements S3	Notifications d'événements Amazon S3
`s3.amazonaws.com`	Inventaire S3	Inventaire Simple Storage Service (Amazon S3)
`access-grants.s3.amazonaws.com`	Octrois d’accès S3	Enregistrement d’un emplacement
`batchoperations.s3.amazonaws.com`	Opérations par lot S3	Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3)
`logging.s3.amazonaws.com`	Journalisation des accès au serveur S3	Activation de la journalisation des accès au serveur Amazon S3
`storage-lens.s3.amazonaws.com`	S3 Storage Lens	Afficher les métriques Amazon S3 Storage Lens à l'aide d'une exportation de données

Politiques et autorisations dans Amazon S3

Cette page présente les stratégies de compartiment et utilisateur dans Amazon S3 et décrit les éléments de base d'une stratégie. Chaque élément répertorié renvoie vers des informations complémentaires sur cet élément et des exemples de son utilisation.

Pour obtenir la liste complète des actions, ressources et conditions d'Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Une stratégie de base contient les éléments suivants :

- Ressource : le compartiment, l'objet, le point d'accès ou la tâche Amazon S3 auxquels s'applique la politique. Utilisez l'Amazon Resource Name (ARN) du compartiment, de l'objet, du point d'accès ou de la tâche pour identifier la ressource.

  Exemple d'opérations au niveau du compartiment :

  - `"Resource": "arn:aws:s3:::bucket_name"`.

  Exemples d'opérations au niveau de l'objet :

  – `"Resource": "arn:aws:s3:::bucket_name/*"` pour tous les objets du compartiment.

  – `"Resource": "arn:aws:s3:::bucket_name/prefix/*"` pour les objets placés sous un certain préfixe dans le compartiment.

  Pour plus d'informations, consultez Ressources relatives aux politiques pour Amazon S3.

- Actions – Pour chaque ressource, Amazon S3 prend en charge un ensemble d'opérations. Vous identifiez les opérations de ressource que vous accordez (ou refusez) en utilisant des mots clés d'action.

  Par exemple, l'autorisation `s3:ListBucket` permet à l'utilisateur d'effectuer l'opération Amazon S3 GET Bucket (List Objects). Pour plus d'informations sur l'utilisation des actions Amazon S3, consultez Actions politiques pour Amazon S3. Pour obtenir la liste complète des actions Amazon S3, consultez Actions.

- Effet – L'effet produit lorsque l'utilisateur demande l'action spécifique, qui peut être un accord ou un refus.

  Si vous n'octroyez pas explicitement l'accès pour (autoriser) une ressource, l'accès est implicitement refusé. Vous pouvez explicitement refuser l'accès à une ressource. Vous pouvez le faire afin de vous assurer qu'un utilisateur n'y a pas accès, même si une stratégie différente accorde cet accès. Pour plus d'informations, consultez Éléments de politique JSON IAM : Effect.

- Principal – Compte ou utilisateur autorisé à accéder aux actions ou aux ressources dans l'instruction. Dans une stratégie de compartiment, le principal est l'utilisateur, le compte, le service ou toute autre entité destinataire de cette autorisation. Pour plus d'informations, consultez Principes relatifs aux politiques relatives aux compartiments.

- Condition – Conditions relatives au moment où une stratégie entre en vigueur. Vous pouvez utiliser AWS des clés larges et des clés spécifiques à Amazon S3 pour spécifier les conditions d'une

politique d'accès Amazon S3. Pour plus d'informations, consultez Exemples de politiques relatives aux compartiments utilisant des clés de condition.

L'exemple de stratégie de compartiment suivant montre les éléments Effect (effet), Principal (mandataire), Action et Resource (ressource). La politique autorise Akua, un utilisateur inscrit dans le compte *Account-ID*,, `s3:GetObjects3:GetBucketLocation`, et `s3:ListBucket` Amazon S3 à accéder au compartiment. `awsexamplebucket1`

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Akua"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:::awsexamplebucket1"
            ]
        }
    ]
}
```

Pour obtenir des informations complètes sur le langage des politiques, consultez les sections Politiques et autorisations dans IAM et Référence des politiques IAM JSON dans le guide de l'utilisateur IAM.

Délégation d'autorisations

Si une personne Compte AWS possède une ressource, elle peut accorder ces autorisations à une autre personne Compte AWS. Ce compte peut alors déléguer à ces utilisateurs, l'ensemble de ces autorisations ou un sous-ensemble de celles-ci. C'est ce que l'on appelle la délégation d'autorisation.

Mais un compte qui reçoit des autorisations d'un autre compte ne peut pas déléguer des autorisations entre comptes à un autre Compte AWS.

Propriété du compartiment et de l'objet Amazon S3

Les compartiments et les objets sont des ressources Amazon S3. Par défaut, seul le propriétaire de ressource peut accéder à ces ressources. Le propriétaire de la ressource fait référence à Compte AWS celui qui crée la ressource. Par exemple :

- Celui Compte AWS que vous utilisez pour créer des buckets et télécharger des objets possède ces ressources.
- Si vous chargez un objet à l'aide des informations d'identification d'utilisateur ou de rôle AWS Identity and Access Management (IAM), l' Compte AWS utilisateur ou le rôle auquel appartient l'utilisateur ou le rôle est propriétaire de l'objet.
- Le propriétaire d'un bucket peut accorder des autorisations entre comptes à un autre Compte AWS (ou à des utilisateurs d'un autre compte) pour le téléchargement d'objets. Dans ce cas, Compte AWS celui qui télécharge les objets est propriétaire de ces objets. Le propriétaire du compartiment n'a aucune autorisation sur les objets dont sont propriétaires d'autres comptes, à l'exception des cas suivants :
  - Le propriétaire du compartiment paie les factures. Le propriétaire du compartiment peut refuser l'accès aux objets ou supprimer des objets dans le compartiment, quel que soit le propriétaire de ces derniers.
  - Le propriétaire du compartiment peut archiver n'importe quel objet ou restaurer des objets archivés, quel que soit le propriétaire de ces derniers. L'archivage fait référence à la classe de stockage utilisée pour stocker les objets. Pour plus d’informations, consultez Gestion du cycle de vie de votre stockage.

Titularité et authentification de demande

Toutes les demandes à un compartiment sont soit authentifiées ou non authentifiées. Les demandes authentifiées doivent inclure une valeur de signature qui authentifie l'expéditeur de la demande, alors que les demandes non authentifiées. Pour plus d'informations sur l'authentification des demandes, veuillez consulter Demandes.

Un propriétaire de compartiment peut choisir d'autoriser les demandes non authentifiées. Par exemple, les PUT Objectdemandes non authentifiées sont autorisées lorsqu'un bucket dispose d'une politique de bucket public, ou lorsqu'une ACL de bucket accorde un `FULL_CONTROL` accès au `All Users` groupe `WRITE` ou à l'utilisateur anonyme en particulier. Pour plus d’informations sur

les politiques de compartiment public et les listes de contrôle d'accès (ACL) publiques, consultez La signification du mot « public ».

Toutes les demandes non authentifiées sont faites par l'utilisateur anonyme. Cet utilisateur est représenté dans les listes ACL par l'ID d'utilisateur canonique spécifique `65a011a29cdf8ec533ec3d1ccaae921c`. Si un objet est chargé sur un compartiment via une demande non authentifiée, l'utilisateur anonyme est propriétaire de l'objet. L'ACL d'objet par défaut autorise le `FULL_CONTROL` pour l'utilisateur anonyme en tant que propriétaire de l'objet. Ainsi, Amazon S3 autorise les demandes non authentifiées pour récupérer l'objet ou modifier son ACL.

Pour empêcher que les objets soient modifiés par un utilisateur anonyme, nous vous recommandons de ne pas implémenter des stratégies de compartiment qui autorisent des écritures publiques anonymes sur votre compartiment ou qui utilisent des ACL pour permettre à l'utilisateur anonyme de disposer d'un accès en écriture à votre compartiment. Vous pouvez faire appliquer ce comportement recommandé à l'aide du blocage de l'accès public Amazon S3.

Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

> **Important**
>
> Nous vous recommandons de ne pas utiliser les informations d'identification de l'utilisateur Compte AWS root pour effectuer des demandes authentifiées. Il est préférable de créer un rôle IAM, puis de lui accorder un accès total. Nous appelons les utilisateurs possédant ce rôle des administrateurs. Vous pouvez utiliser les informations d'identification attribuées au rôle d'administrateur, au lieu des informations d'identification de l'utilisateur Compte AWS root, pour interagir avec AWS et effectuer des tâches, telles que créer un bucket, créer des utilisateurs et accorder des autorisations. Pour plus d'informations, consultez les informations d'identification AWS de sécurité dans le guide de l'utilisateur IAM et les meilleures pratiques de sécurité dans IAM dans le guide de l'utilisateur IAM.

Politiques relatives aux compartiments pour Amazon S3

Une politique de compartiment est une politique basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment Amazon S3 et aux objets qu'il

contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Ces autorisations ne s'appliquent pas aux objets appartenant à d'autres personnes Comptes AWS.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes de contrôle d'accès (ACL). Par défaut, la propriété d'objets est définie sur le paramètre Propriétaire du compartiment imposé, et toutes les listes ACL sont désactivées. Le propriétaire du compartiment détient tous les objets présents dans le compartiment et gère l'accès aux données exclusivement au moyen de politiques.

Les politiques de compartiment utilisent un langage de stratégie basé sur JSON AWS Identity and Access Management (IAM). Vous pouvez utiliser des stratégies de compartiment pour ajouter ou refuser des autorisations pour les objets d'un compartiment. Les politiques de compartiment peuvent autoriser ou refuser les requêtes en fonction des éléments de la politique. Ces éléments comprennent le demandeur, les actions S3, les ressources et les aspects ou conditions de la requête (comme l'adresse IP utilisée pour effectuer la requête).

Par exemple, vous pouvez créer une politique de compartiment qui effectue les actions suivantes :

- Accorde à d'autres comptes des autorisations intercompte pour charger des objets dans votre compartiment S3.
- Assurez-vous que vous, le propriétaire du compartiment, avez le contrôle total des objets chargés.

Pour plus d'informations, consultez Exemples de politiques relatives aux compartiments Amazon S3.

> **Important**
>
> Vous ne pouvez pas utiliser une politique de compartiment pour empêcher les suppressions ou les transitions selon une règle du cycle de vie S3. Par exemple, même si votre politique de compartiment refuse toutes les actions pour tous les principaux, votre configuration S3 Lifecycle fonctionne toujours normalement.

Les rubriques de cette section fournissent des exemples et indiquent comment ajouter une stratégie de compartiment dans la console S3. Pour plus d'informations sur les politiques basées sur l'identité,

consultez. Politiques basées sur l'identité pour Amazon S3 Pour en savoir plus sur le langage des stratégies de compartiment, consultez Politiques et autorisations dans Amazon S3.

Rubriques

- Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3
- Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment
- Exemples de politiques relatives aux compartiments Amazon S3
- Exemples de politiques relatives aux compartiments utilisant des clés de condition

Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3

Vous pouvez utiliser le générateur de politiques AWS et la console Amazon S3 pour ajouter une nouvelle politique de compartiment ou modifier une politique de compartiment existante. Une politique de compartiment est une politique basée sur les ressources AWS Identity and Access Management (IAM). Vous ajoutez une politique de compartiment à un compartiment pour accorder à d'autres utilisateurs Comptes AWS ou à des utilisateurs IAM des autorisations d'accès pour le compartiment et les objets qu'il contient. Les autorisations d'objets ne s'appliquent qu'aux objets créés par le propriétaire du compartiment. Pour plus d'informations sur les stratégies de compartiment, consultez Identity and Access Management pour Amazon S3.

Veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions provenant d' AWS Identity and Access Management Access Analyzer avant d'enregistrer votre stratégie. IAM Access Analyzer exécute des vérifications de politiques pour valider votre politique par rapport à la grammaire de politique et aux bonnes pratiques IAM. Ces vérifications génèrent des résultats et fournissent des recommandations exploitables pour vous aider à créer des stratégies fonctionnelles et conformes aux bonnes pratiques en matière de sécurité. Pour en savoir plus sur la validation des politiques à l'aide d'IAM Access Analyzer, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM. Pour afficher la liste des avertissements, erreurs et suggestions renvoyés par IAM Access Analyzer, consultez la Référence de vérification de stratégie IAM Access Analyzer.

Pour obtenir des conseils sur la résolution des erreurs liées à une politique, consultez Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3.

Pour créer ou modifier une stratégie de compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez créer une stratégie de compartiment ou modifier la stratégie de compartiment existante.
4. Choisissez l'onglet Permissions (Autorisations).
5. Sous Politique de compartiment, choisissez Modifier. La page Edit bucket policy (Modifier la politique de compartiment) s'affiche.
6. Dans la page Edit bucket policy (Modifier la politique de compartiment), procédez de l'une des manières suivantes :

   - Pour voir des exemples de stratégies de compartiment dans le Guide de l'utilisateur d'Amazon S3, sélectionnez Policy examples (Exemples de politiques).
   - Pour générer une politique automatiquement, ou modifier le JSON dans la section Policy (Politique), choisissez Policy generator (Générateur de politique).

   Si vous choisissez le générateur de AWS politiques, celui-ci s'ouvre dans une nouvelle fenêtre.

   a. Sur la page AWS Policy Generator (Générateur de politiques), pour Select Type of Policy (Sélectionner le type de politique), sélectionnez S3 Bucket Policy (Politique de compartiment S3).

   b. Ajoutez une instruction en saisissant les informations dans les champs fournis, puis choisissez Add Statement (Ajouter une instruction). Répétez l'opération pour autant d'instructions que vous souhaitez ajouter. Pour plus d'informations sur ces champs, consultez la Référence des éléments de stratégie IAM JSON dans le Guide de l'utilisateur IAM.

   > **Note**
   >
   > Pour plus de commodité, la page Edit Bucket Policy (Modifier la politique de compartiment) affiche l'ARN (Amazon Resource Name) de compartiment du compartiment actuel au-dessus du champ de texte Policy (Politique). Vous pouvez

copier cet ARN pour l'utiliser dans les instructions de la page AWS Policy Generator (Générateur de politique).

c. Une fois que vous avez fini d'ajouter des instructions, choisissez Generate Policy (Générer une stratégie).

d. Copiez le texte de stratégie généré, choisissez Close (Fermer) et revenez à la page Edit bucket policy (Modifier la stratégie de compartiment) dans la console Amazon S3.

7. Dans la zone Politique, modifiez la politique existante ou collez la politique de compartiment depuis le générateur de AWS politiques. Veillez à résoudre les avertissements de sécurité, les erreurs, les avertissements généraux et les suggestions avant d'enregistrer votre stratégie.

> **Note**
>
> Les stratégies de compartiment sont limitées à une taille de 20 Ko.

8. (Facultatif) Choisissez Preview external access (Aperçu de l'accès externe) dans le coin inférieur droit pour avoir un aperçu de la façon dont votre nouvelle politique affecte l'accès public et l'accès intercompte à votre ressource. Avant d'enregistrer votre stratégie, vous pouvez vérifier si elle introduit de nouveaux résultats IAM Access Analyzer ou si elle résout les résultats existants. Si vous ne voyez pas d'analyseur actif, choisissez Go to Access Analyzer (Accédez à l'analyseur d'accès) pour créer un analyseur de compte dans l'analyseur d'accès IAM. Pour plus d'informations, consultez Prévisualisation des accès dans le Guide de l'utilisateur IAM.
9. Choisissez Save changes (Enregistrer les modifications), ce qui vous ramène à l'onglet Permissions (Autorisations).

Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment

Vous pouvez utiliser les politiques relatives aux compartiments Amazon S3 pour contrôler l'accès aux compartiments à partir de points de terminaison de cloud privé virtuel (VPC) ou de VPC spécifiques. Cette section contient des exemples de politiques de compartiment que vous pouvez utiliser pour contrôler l'accès aux compartiments Amazon S3 à partir des points de terminaison VPC. Pour apprendre à configurer les points de terminaison d'un VPC, veuillez consulter Points de terminaison d'un VPC dans le Guide de l'utilisateur VPC.

Un VPC vous permet de lancer des AWS ressources dans un réseau virtuel que vous définissez. Un point de terminaison VPC vous permet de créer une connexion privée entre votre VPC et un autre.

Service AWS Cette connexion privée ne nécessite pas d'accès via Internet, via une connexion de réseau privé virtuel (VPN), via une instance NAT ou via AWS Direct Connect.

Le point de terminaison d'un VPC pour Amazon S3 est une entité logique au sein d'un VPC qui permet uniquement une connexion à Amazon S3. Le point de terminaison d'un VPC achemine les demandes vers Amazon S3 et les réponses renvoyées au VPC. Les points de terminaison d'un VPC changent uniquement la manière dont les demandes sont acheminées . Les points de terminaison publics Amazon S3 et les noms DNS continuent de fonctionner avec les points de terminaison d'un VPC. Pour obtenir des informations importantes sur l'utilisation des points de terminaison VPC avec Amazon S3, consultez la section Points de terminaison de passerelle et points de terminaison de passerelle pour Amazon S3 dans le guide de l'utilisateur VPC.

Les points de terminaison d'un VPC pour Amazon S3 offrent deux façons de contrôler l'accès à vos données Amazon S3 :

- Vous pouvez contrôler les demandes, les utilisateurs ou les groupes autorisés à traverser un point de terminaison d'un VPC spécifique. Pour plus d'informations sur ce type de contrôle d'accès, consultez la section Contrôle de l'accès aux points de terminaison VPC à l'aide de politiques relatives aux points de terminaison dans le Guide de l'utilisateur VPC.
- Vous pouvez contrôler quels VPC ou points de terminaison d'un VPC ont accès à vos compartiments en utilisant des stratégies de compartiment Amazon S3. Pour obtenir des exemples de ce type de contrôle d'accès avec stratégie de compartiment, consultez les rubriques suivantes sur les restrictions d'accès.

Rubriques

- Restriction de l'accès à un point de terminaison d'un VPC spécifique
- Restriction de l'accès à un VPC spécifique

> **Important**
>
> Lorsque vous appliquez les politiques de compartiment Amazon S3 pour les points de terminaison VPC décrites dans cette section, vous risquez de bloquer involontairement votre accès au compartiment. Les autorisations attribuées à un compartiment dans le but de restreindre l'accès aux connexions issues du point de terminaison de votre VPC peuvent bloquer toutes les connexions à ce compartiment. Pour plus d'informations sur la manière de résoudre ce problème, consultez Comment corriger ma politique de compartiment lorsque le

VPC ou l'ID de point de terminaison du VPC est incorrect ? dans le AWS Support Knowledge Center.

Restriction de l'accès à un point de terminaison d'un VPC spécifique

Voici un exemple de stratégie de compartiment Amazon S3 qui restreint l'accès à un compartiment spécifique, `awsexamplebucket1`, uniquement à partir du point de terminaison d'un VPC doté de l'ID `vpce-1a2b3c4d`. Si le point de terminaison spécifié n'est pas utilisé, la politique refuse tout accès au compartiment. La `aws:SourceVpce` condition spécifie le point de terminaison. La `aws:SourceVpce` condition ne nécessite pas de nom de ressource Amazon (ARN) pour la ressource de point de terminaison du VPC, mais uniquement l'ID du point de terminaison du VPC. Pour plus d'informations sur l'utilisation de conditions dans une stratégie, consultez Exemples de politiques relatives aux compartiments utilisant des clés de condition.

Important

- Avant d'utiliser l'exemple de stratégie suivant, remplacez l'ID de point de terminaison du VPC par une valeur appropriée pour votre cas d'utilisation. Dans le cas contraire, vous ne parviendrez pas à accéder à votre compartiment.
- Cette politique désactive l'accès de la console au compartiment spécifié car les demandes de console ne proviennent pas du point de terminaison VPC spécifié.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909152",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPCE-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
       "Resource": ["arn:aws:s3:::awsexamplebucket1",
                    "arn:aws:s3:::awsexamplebucket1/*"],
       "Condition": {
         "StringNotEquals": {
           "aws:SourceVpce": "vpce-1a2b3c4d"
         }
```

```
        }
      }
    ]
}
```

Restriction de l'accès à un VPC spécifique

Vous pouvez créer une stratégie de compartiment qui restreint l'accès à un VPC spécifique en utilisant la condition `aws:SourceVpc`. Ceci est utile si vous avez plusieurs points de terminaison d'un VPC configurés pour le même VPC et que vous voulez gérer l'accès à vos compartiments Amazon S3 pour tous vos points de terminaison. Voici un exemple de politique qui refuse l'accès à `awsexamplebucket1` et ses objets à toute personne extérieure au VPC `vpc-111bbb22`. Si le VPC spécifié n'est pas utilisé, la politique refuse tout accès au compartiment. Cette instruction n'autorise pas l'accès au bucket. Pour autoriser l'accès, vous devez ajouter une `Allow` déclaration séparée. La clé de `vpc-111bbb22` condition ne nécessite pas d'ARN pour la ressource VPC, uniquement l'ID du VPC.

> **Important**
>
> - Avant d'utiliser l'exemple de stratégie suivant, remplacez l'ID du VPC par une valeur appropriée pour votre cas d'utilisation. Dans le cas contraire, vous ne parviendrez pas à accéder à votre compartiment.
> - Cette politique désactive l'accès de la console au compartiment spécifié car les demandes de console ne proviennent pas du VPC spécifié.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909153",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPC-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
       "Resource": ["arn:aws:s3:::awsexamplebucket1",
                    "arn:aws:s3:::awsexamplebucket1/*"],
       "Condition": {
         "StringNotEquals": {
```

```
            "aws:SourceVpc": "vpc-111bbb22"
          }
        }
      }
    ]
}
```

Exemples de politiques relatives aux compartiments Amazon S3

Grâce aux politiques de compartiment d'Amazon S3, vous pouvez sécuriser l'accès aux objets de vos compartiments, afin que seuls les utilisateurs disposant des autorisations appropriées puissent y accéder. Vous pouvez même empêcher les utilisateurs authentifiés ne disposant pas des autorisations appropriées d'accéder à vos ressources Amazon S3.

Cette section présente des exemples de cas d'utilisation standard de politiques de compartiment. Ces exemples de politiques utilisent *example-s3-bucket* comme valeur de ressource. Pour tester ces politiques, remplacez *user input placeholders* par vos propres informations (comme le nom de votre compartiment).

Pour accorder ou refuser des autorisations à un ensemble d'objets, vous pouvez utiliser des caractères génériques (*) dans les noms Amazon Resource Name (ARN) et d'autres valeurs. Par exemple, vous pouvez contrôler l'accès à des groupes d'objets qui commencent par un préfixe commun ou se terminent par une extension spécifique, tels que `.html`.

Pour plus d'informations sur le langage de politique AWS Identity and Access Management (IAM), consultezPolitiques et autorisations dans Amazon S3.

> **Note**
>
> Lors du test des autorisations à l'aide de la console Amazon S3, vous devez accorder les autorisations supplémentaires requises par la console (`s3:ListAllMyBuckets`, `s3:GetBucketLocation` et `s3:ListBucket`). Pour obtenir un exemple de guide étape par étape pour accorder des autorisations aux utilisateurs et tester ces autorisations à l'aide de la console, consultez Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur.

Les ressources supplémentaires pour créer des politiques de compartiment sont les suivantes :

- Pour obtenir la liste des actions de politique IAM, des ressources et des clés de condition que vous pouvez utiliser lors de la création d'une politique de compartiment, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference.
- Pour obtenir des conseils sur la création de votre politique S3, consultez Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3.
- Pour résoudre les erreurs liées à une politique, consultez Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3.

Rubriques

Octroi d'une autorisation de lecture seule à un utilisateur public anonyme

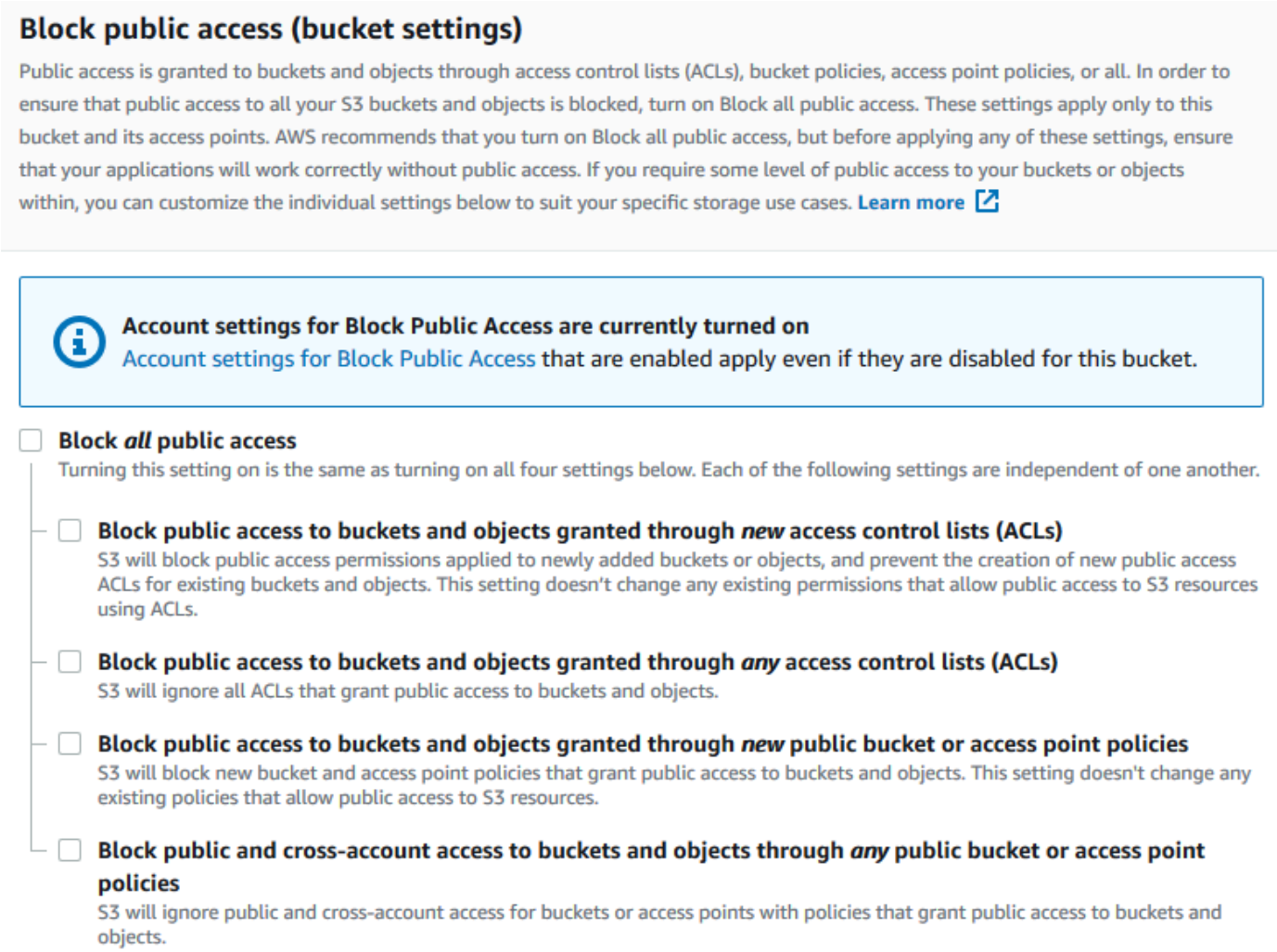
Vous pouvez utiliser vos paramètres de politique pour accorder l'accès à des utilisateurs anonymes publics, ce qui est utile si vous configurez votre bucket en tant que site Web statique. Cela nécessite que vous désactiviez le blocage de l'accès public pour votre compartiment. Pour plus d'informations sur la procédure à suivre et sur la politique requise, consultezDéfinition des autorisations pour l'accès au site web. Pour savoir comment configurer des politiques plus restrictives dans le même but, consultez Comment accorder un accès public en lecture à certains objets de mon compartiment Amazon S3 ? dans le AWS Knowledge Center.

Par défaut, Amazon S3 bloque l'accès public à votre compte et à vos compartiments. Si vous souhaitez utiliser un compartiment pour héberger un site web statique, vous pouvez utiliser ces étapes pour modifier vos paramètres de blocage de l'accès public.

> **Warning**
>
> Avant de terminer cette étape, revoyez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tout accès public à vos compartiments.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment que vous avez configuré en tant que site web statique.
3. Choisissez Permissions.
4. Sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)), choisissez Edit (Modifier).
5. Effacez Block all public access (Bloquer tous les accès publics) et choisissez Enregistrer les modifications.

> **Warning**
>
> Avant de terminer cette étape, examinez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tous les accès publics à vos compartiments.



Amazon S3 désactive les paramètres de blocage de l'accès public pour votre compartiment. Pour créer un site web public statique, vous devrez peut-être aussi modifier les paramètres de blocage de l'accès public de votre compte avant d'ajouter une stratégie de compartiment. Si les paramètres du compte pour la fonctionnalité de blocage de l'accès public sont actuellement activés, une note s'affiche sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)).

Obligation de chiffrement

Exiger le SSE-KMS pour tous les objets enregistrés dans un compartiment

L'exemple de politique suivant exige que chaque objet écrit dans le compartiment soit chiffré avec un chiffrement côté serveur à l'aide de clés AWS Key Management Service (AWS KMS) (SSE-KMS). Si l'objet n'est pas chiffré avec SSE-KMS, la demande est refusée.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMS",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
    }
  }
}]
}
```

Exiger un SSE-KMS avec un AWS KMS key spécifique pour tous les objets enregistrés dans un compartiment

L'exemple de politique suivant interdit l'écriture d'objets dans le compartiment s'ils ne sont pas chiffrés avec SSE-KMS en utilisant un ID de clé KMS spécifique. Même si les objets sont chiffrés avec SSE-KMS à l'aide d'un en-tête par demande ou d'un chiffrement par défaut du compartiment, les objets ne peuvent pas être écrits dans le compartiment s'ils n'ont pas été chiffrés avec la clé KMS spécifiée. Assurez-vous de remplacer la clé KMS ARN utilisée dans cet exemple par votre propre clé KMS ARN.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
  "Principal": "*",
  "Effect": "Deny",
```

```
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
  "Condition": {
    "ArnNotEqualsIfExists": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
    }
  }
}]
}
```

Gestion des compartiments à l'aide de listes de contrôle d'accès en prédéfinies (ACL)

Accorder des autorisations à plusieurs comptes pour charger des objets ou définir des listes de contrôle d'accès aux objets pour l'accès public.

L'exemple de politique suivant accorde les `s3:PutObjectAcl` autorisations `s3:PutObject` et à plusieurs Comptes AWS. En outre, l'exemple de politique exige que toutes les demandes relatives à ces opérations incluent la liste de `public-read` contrôle d'accès (ACL) prédéfinie. Pour plus d'informations, consultez Actions politiques pour Amazon S3 et Clés de conditions de politique pour Amazon S3.

> **Warning**
>
> L'ACL prédéfinie `public-read` permet à n'importe qui dans le monde entier de visualiser les objets de votre compartiment. Soyez vigilant lors de l'octroi de l'accès anonyme à votre compartiment Amazon S3 ou de la désactivation des paramètres du blocage de l'accès public. Lorsque vous accordez un accès anonyme, tout le monde peut accéder à votre compartiment. Il est recommandé de ne jamais autoriser un accès anonyme à votre compartiment Amazon S3 à moins que vous n'en ayez spécifiquement besoin, comme dans le cas de l'hébergement de site web statique. Si vous souhaitez activer les paramètres de blocage de l'accès public pour l'hébergement de sites Web statiques, consultez Tutoriel : configuration d'un site web statique sur Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AddPublicReadCannedAcl",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root"
                ]
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ]
                }
            }
        }
    ]
}
```

Octroi d'autorisations intercomptes pour charger des objets tout en garantissant que le propriétaire du compartiment dispose d'un contrôle total

L'exemple suivant montre comment autoriser une autre personne Compte AWS à télécharger des objets dans votre compartiment tout en vous assurant que vous avez le contrôle total des objets chargés. Cette politique accorde à un compte spécifique Compte AWS (*111122223333*) la possibilité de télécharger des objets uniquement si ce compte inclut l'ACL `bucket-owner-full-control` prédéfinie lors du téléchargement. La condition `StringEquals` figurant dans la politique spécifie la clé de la condition `s3:x-amz-acl` pour exprimer l'exigence de l'ACL prédéfinie. Pour plus d'informations, consultez Clés de conditions de politique pour Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
     {
       "Sid":"PolicyForAllowUploadWithACL",
       "Effect":"Allow",
       "Principal":{"AWS":"111122223333"},
       "Action":"s3:PutObject",
```

```
        "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
        "Condition": {
          "StringEquals": {"s3:x-amz-acl":"bucket-owner-full-control"}
        }
      }
   ]
}
```

Gestion de l'accès aux objets avec l'étiquetage des objets

Autoriser un utilisateur à lire uniquement les objets qui ont une clé et une valeur d'étiquette spécifiques

La politique d'autorisations suivante limite un utilisateur à la seule lecture des objets qui comportent la clé et la valeur d'étiquette `environment: production`. Cette politique utilise la clé de condition `s3:ExistingObjectTag` pour spécifier la clé et la valeur d'étiquette.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Principal":{
            "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
         },
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/environment":"production"
            }
         }
      }
   ]
}
```

Restreindre les clés d'étiquette d'objet que les utilisateurs peuvent ajouter

L'exemple de politique suivant accorde à un utilisateur l'autorisation d'exécuter l'action `s3:PutObjectTagging`, qui permet à un utilisateur d'ajouter des étiquettes à un objet existant. La

condition utilise la clé de condition `s3:RequestObjectTagKeys` pour spécifier les clés d'étiquette autorisées, telles que `Owner` ou `CreationDate`. Pour plus d'informations, consultez Création d'une condition avec plusieurs clés ou valeurs dans le Guide de l'utilisateur IAM.

Cette politique garantit que chaque clé d'étiquette spécifiée dans la demande est une clé d'étiquette autorisée. Le qualificateur `ForAnyValue` dans la condition garantit qu'au moins une des clés spécifiées doit être présente dans la demande.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Exiger une clé et une valeur d'étiquette spécifiques pour permettre aux utilisateurs d'ajouter des étiquettes d'objet

L'exemple de politique suivant accorde à un utilisateur l'autorisation d'exécuter l'action `s3:PutObjectTagging`, qui permet à un utilisateur d'ajouter des étiquettes à un objet existant. La condition exige que l'utilisateur inclue une clé d'étiquette spécifique (telle que *`Project`*) avec la valeur définie sur *`X`*.

```
{
```

```
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
        ]
       },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Permettre à un utilisateur de n'ajouter que des objets ayant une clé et une valeur d'étiquetage d'objet spécifiques

L'exemple de politique suivant accorde à un utilisateur l'autorisation d'effectuer l'action `s3:PutObject` afin qu'il puisse ajouter des objets à un compartiment. Cependant, l'instruction `Condition` restreint les clés et les valeurs d'étiquetage qui sont autorisées sur les objets chargés. Dans cet exemple, l'utilisateur ne peut ajouter au compartiment que les objets ayant la clé d'étiquette spécifique (*`Department`*) avec la valeur définie sur *`Finance`*.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Principal":{
            "AWS":[
                 "arn:aws:iam::111122223333:user/JohnDoe"
         ]
        },
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
        ],
        "Condition": {
            "StringEquals": {
                "s3:RequestObjectTag/Department": "Finance"
            }
        }
    }]
}
```

Gestion de l'accès aux objets par l'utilisation de clés de condition globales

Les clés de condition globales sont des clés de contexte de condition avec un `aws` préfixe. Services AWS peut prendre en charge les clés de condition globales ou les clés spécifiques au service qui incluent le préfixe du service. Vous pouvez utiliser l'élément `Condition` d'une politique JSON pour comparer les clés d'une requête avec les valeurs des clés que vous spécifiez dans votre politique.

Restreindre l'accès aux seules livraisons du journal d'accès du serveur Amazon S3

Dans l'exemple de politique de compartiment suivant, la clé de condition aws:SourceArnglobale est utilisée pour comparer le nom de ressource Amazon (ARN) de la ressource, en effectuant une service-to-service demande avec l'ARN spécifié dans la politique. La clé de condition globale `aws:SourceArn` est utilisée pour empêcher le service Amazon S3 d'être utilisé comme adjoint confus lors de transactions entre services. Seul le service Amazon S3 est autorisé à ajouter des objets au compartiment Amazon S3.

Cet exemple de stratégie de compartiment accorde uniquement des autorisations `s3:PutObject` au principal du service de journalisation (`logging.s3.amazonaws.com`).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-logs/*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:SourceAccount": "111111111111"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
                }
            }
        },
        {
            "Sid": "RestrictToS3ServerAccessLogs",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET-logs/*",
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "aws:PrincipalServiceNamesList": "logging.s3.amazonaws.com"
                }
            }
        }
    ]
}
```

Autoriser l'accès uniquement à votre organisation

Si vous souhaitez que tous les principaux IAM accédant à une ressource proviennent d'un Compte AWS compte de gestion (y compris le compte de AWS Organizations gestion), vous pouvez utiliser la clé de condition globale `aws:PrincipalOrgID`.

Pour accorder ou restreindre ce type d'accès, définissez la condition `aws:PrincipalOrgID` et définissez la valeur de l'ID de votre organisation dans la politique de compartiment. L'ID de l'organisation permet de contrôler l'accès au compartiment. Lorsque vous utilisez la condition `aws:PrincipalOrgID`, les autorisations de la politique de compartiment sont également appliquées à tous les nouveaux comptes qui sont ajoutés à l'organisation.

Voici un exemple de politique de compartiment basée sur les ressources que vous pouvez utiliser pour accorder à des principaux IAM spécifiques de votre organisation un accès direct à votre compartiment. En ajoutant la clé de condition globale `aws:PrincipalOrgID` à votre politique de compartiment, le compte principal doit désormais faire partie de votre organisation pour obtenir l'accès à la ressource. Même si vous spécifiez accidentellement un compte incorrect lors de l'octroi de l'accès, la clé de condition globale aws:PrincipalOrgID constitue une protection supplémentaire. Lorsque cette clé globale est utilisée dans une politique, elle empêche tous les principaux de

l'extérieur de l'organisation spécifiée d'accéder au compartiment S3. Seuls les principaux des comptes de l'organisation listée peuvent obtenir l'accès à la ressource.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowGetObject",
        "Principal": {
            "AWS": "*"
        },
        "Effect": "Allow",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": ["o-aa111bb222"]
            }
        }
    }]
}
```

Gestion de l'accès en fonction d'adresses IP spécifiques

Restriction de l'accès à des adresses IP spécifiques

L'exemple suivant interdit à tous les utilisateurs d'effectuer des opérations Amazon S3 sur les objets des compartiments spécifiés, sauf si la requête provient de la plage d'adresses IP spécifiée.

> **Note**
>
> Lorsque vous limitez l'accès à une adresse IP spécifique, assurez-vous de spécifier également les points de terminaison de VPC, adresses IP source de VPC ou adresses IP externes qui peuvent accéder au compartiment S3. Dans le cas contraire, vous risquez de perdre l'accès au compartiment si votre politique interdit à tous les utilisateurs d'effectuer des opérations S3 sur les objets de votre compartiment sans que les autorisations appropriées ne soient déjà en place.

Cette instruction de `Condition` de politique identifie *`192.0.2.0/24`* comme la plage d'adresses Internet Protocol version 4 (IPv4) autorisées.

Le `Condition` bloc utilise la `NotIpAddress` condition et la clé de `aws:SourceIp` condition, qui est une clé AWS de condition large. La clé de condition `aws:SourceIp` ne peut être utilisée que pour les plages d'adresses IP publiques. Pour plus d'informations sur l'utilisation de ces clés de condition, consultez Clés de conditions de politique pour Amazon S3. Les valeurs IPv4 `aws:SourceIp` font appel à la notation CIDR standard. Pour plus d'informations, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

> **Warning**
>
> Avant d'employer cette politique, remplacez la plage d'adresses IP *192.0.2.0/24* dans cet exemple par une valeur appropriée pour votre cas d'utilisation. Dans le cas contraire, vous perdrez la possibilité d'accéder à votre compartiment.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                }
            }
        }
    ]
}
```

Autoriser les adresses IPv4 et IPv6

Lorsque vous commencez à utiliser des adresses IPv6, nous vous recommandons de mettre à jour toutes les stratégies de votre organisation en y incluant des plages d'adresses IPv6, en plus des

plages IPv4 existantes. Cela permettra de s'assurer que les politiques continuent de fonctionner lors de la transition vers IPv6.

L'exemple de stratégie de compartiment ci-dessous montre comment combiner des plages d'adresses IPv4 et IPv6 pour couvrir la totalité des adresses IP valides de l'organisation. Dans cet exemple, la politique autorise l'accès aux exemples d'adresses IP *192.0.2.1* et *2001:DB8:1234:5678::1* et le refuse aux adresses *203.0.113.1* et *2001:DB8:1234:5678:ABCD::1*.

La clé de condition `aws:SourceIp` ne peut être utilisée que pour les plages d'adresses IP publiques. Les valeurs IPv6 pour `aws:SourceIp` doivent être au format CIDR standard. Pour IPv6, nous prenons en charge l'utilisation de `::` pour représenter une plage de zéros (par exemple : `2001:DB8:1234:5678::/64`). Pour plus d'informations, consultez Opérateurs de condition d'adresse IP dans le Guide de l'utilisateur IAM.

> **Warning**
>
> Remplacez les plages d'adresses IP de cet exemple par des valeurs appropriées pour votre cas d'utilisation avant d'employer cette stratégie. Dans le cas contraire, vous pourriez perdre la possibilité d'accéder à votre compartiment.

```
{
    "Id": "PolicyId2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIPmix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
```

```
                },
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
                        "2001:DB8:1234:5678:ABCD::/80"
                    ]
                }
            }
        }
    ]
}
```

Gestion des accès en fonction des requêtes HTTP ou HTTPS

Restreindre l'accès aux seules requêtes HTTPS

Si vous voulez empêcher les attaquants potentiels de manipuler le trafic réseau, vous pouvez utiliser HTTPS (TLS) pour n'autoriser que les connexions chiffrées tout en limitant l'accès à votre compartiment aux requêtes HTTP. Pour déterminer si la requête est HTTP ou HTTPS, utilisez la clé de condition globale aws:SecureTransport dans votre politique de compartiment S3. La clé de condition `aws:SecureTransport` vérifie si une requête a été envoyée en utilisant HTTP.

Si une demande renvoie `true`, alors la demande a été envoyée par HTTPS. Si une demande renvoie `false`, alors la demande a été envoyée par HTTP. Vous pouvez ensuite autoriser ou refuser l'accès à votre compartiment en fonction du schéma de requête souhaité.

Dans l'exemple suivant, la politique de compartiment refuse explicitement les demandes HTTP.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:*",
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
```

```
        "Principal": "*"
    }]
}
```

Restrindre l'accès à un référent HTTP spécifique

Supposons que vous ayez un site web avec le nom de domaine *www.example.com* ou *example.com*, avec des liens vers des photos et des vidéos stockées dans votre compartiment nommé *example-s3-bucket*. Par défaut, toutes les ressources Amazon S3 sont privées, de sorte que seul Compte AWS celui qui les a créées peut y accéder.

Pour autoriser l'accès en lecture à ces objets à partir de votre site Web, vous pouvez ajouter une politique de compartiment qui accorde l'autorisation `s3:GetObject` avec une condition stipulant que la requête GET provienne initialement de pages Web spécifiques. La politique suivante limite les requêtes en utilisant la condition`StringLike` avec la clé de condition `aws:Referer`.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
 example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::example-s3-bucket/*",
      "Condition":{
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}
      }
    }
  ]
}
```

Veillez à ce que les navigateurs que vous utilisez incluent l'en-tête HTTP `referer` dans la demande.

> **Warning**
>
> Nous vous recommandons de faire preuve de prudence lorsque vous utilisez la clé de condition `aws:Referer`. Il est dangereux d'inclure une valeur d'en-tête de référent HTTP

connu publiquement. Les tiers non autorisés peuvent utiliser des navigateurs modifiés ou personnalisés pour fournir n'importe quelle valeur `aws:Referer` de leur choix. Par conséquent, ne l'utilisez pas `aws:Referer` pour empêcher des parties non autorisées de faire des AWS demandes directes.
La clé de condition `aws:Referer` est fournie uniquement pour permettre aux clients de protéger leur contenu numérique, stocké notamment dans Simple Storage Service (Amazon S3), contre tout référencement sur des sites tiers non autorisés. Pour plus d'informations, consultez aws:Referer dans le Guide de l'utilisateur IAM.

Gestion de l'accès des utilisateurs à des dossiers spécifiques

Accorder aux utilisateurs l'accès à des dossiers spécifiques

Supposons que vous essayez d'accorder aux utilisateurs l'accès à un dossier spécifique. Si l'utilisateur IAM et le compartiment S3 appartiennent à la même entité Compte AWS, vous pouvez utiliser une politique IAM pour accorder à l'utilisateur l'accès à un dossier de compartiment spécifique. Avec cette approche, vous n'avez pas besoin de mettre à jour votre politique de compartiment pour octroyer l'accès. Vous pouvez ajouter la politique IAM à un rôle IAM auquel plusieurs utilisateurs peuvent basculer.

Si l'identité IAM et le compartiment S3 appartiennent à des entités différentes Comptes AWS, vous devez accorder un accès entre comptes à la fois dans la politique IAM et dans la politique de compartiment. Pour plus d'informations sur l'octroi d'un accès intercompte, consultez Bucket owner granting cross-account bucket permissions (Propriétaire du compartiment accordant des autorisations intercomptes pour le compartiment).

L'exemple suivant de politique de compartiment accorde à *JohnDoe* un accès complet à la console à son seul dossier (`home/`*JohnDoe*`/`). En créant un dossier `home` et en accordant les autorisations appropriées à vos utilisateurs, vous pouvez faire en sorte que plusieurs utilisateurs partagent un seul compartiment. Cette politique se compose de trois instructions `Allow` :

- *AllowRootAndHomeListingOfCompanyBucket* : permet à l'utilisateur (*JohnDoe*) de lister les objets au niveau de la racine du compartiment *DOC-EXAMPLE-BUCKET* et dans le dossier `home`. Cette instruction permet également à l'utilisateur d'effectuer une recherche sur le préfixe `home/` en utilisant la console.
- *AllowListingOfUserFolder* : permet à l'utilisateur (*JohnDoe*) de lister tous les objets du dossier `home/`*JohnDoe*`/` et de ses sous-dossiers éventuels.

- *AllowAllS3ActionsInUserFolder* : permet à l'utilisateur d'effectuer toutes les actions Amazon S3 en accordant des autorisations `Read`, `Write` et `Delete`. Les autorisations sont limitées au dossier personnel du propriétaire du compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRootAndHomeListingOfCompanyBucket",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": ["", "home/", "home/JohnDoe"],
                    "s3:delimiter": ["/"]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:ListBucket"],
            "Effect": "Allow",
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"],
            "Condition": {
                "StringLike": {
                    "s3:prefix": ["home/JohnDoe/*"]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:*"],
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/home/JohnDoe/*"]
        }
    ]
}
```

Gestion des accès pour les journaux d'accès

Accorder l'accès à l'Application Load Balancer pour activer les journaux d'accès

Lorsque vous activez les journaux d'accès pour Application Load Balancer, vous devez spécifier le nom du compartiment S3 où l'équilibreur de charge stockera les journaux. Le compartiment doit comporter une politique attachée qui accorde à Elastic Load Balancing l'autorisation d'écrire dans le compartiment.

Dans l'exemple suivant, la politique de compartiment accorde à Elastic Load Balancing (ELB) l'autorisation d'écrire les journaux d'accès dans le compartiment :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

> **Note**
>
> Assurez-vous de remplacer *elb-account-id* par l'ID du Compte AWS d'Elastic Load Balancing pour votre Région AWS. Pour obtenir la liste des régions Elastic Load Balancing,

consultez Attach a policy to your Amazon S3 bucket (Association d'une politique à votre compartiment Amazon S3) dans le Guide de l'utilisateur Elastic Load Balancing.

Si vous Région AWS ne figurez pas dans la liste des régions Elastic Load Balancing prises en charge, appliquez la politique suivante, qui accorde des autorisations au service de livraison de journaux spécifié.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
       "Principal": {
         "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
          },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::example-s3-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Ensuite, veillez à configurer vos journaux d'accès Elastic Load Balancing en les activant. Vous pouvez vérifier les autorisations de votre compartiment en créant un fichier de test.

Gestion de l'accès à un Amazon CloudFront OAI

Accorder une autorisation à un Amazon CloudFront OAI

L'exemple de politique de compartiment suivant accorde à une identité CloudFront d'accès d'origine (OAI) l'autorisation d'obtenir (lire) tous les objets de votre compartiment S3. Vous pouvez utiliser un CloudFront OAI pour autoriser les utilisateurs à accéder aux objets de votre compartiment via Amazon S3 CloudFront , mais pas directement. Pour plus d'informations, consultez Restreindre l'accès au contenu Amazon S3 à l'aide d'une identité d'accès d'origine dans le manuel Amazon CloudFront Developer Guide.

La stratégie suivante utilise l'ID de l'OAI comme stratégie de `Principal`. Pour plus d'informations sur l'utilisation des politiques de compartiment S3 pour accorder l'accès à un CloudFront OAI, consultez la section Migration de l'identité d'accès d'origine (OAI) vers le contrôle d'accès d'origine (OAC) dans le manuel Amazon Developer Guide. CloudFront

Pour utiliser cet exemple :

- Remplacez *EH1HDMB1FH2TC* par l'ID OAI. Pour trouver l'ID de l'OAI, consultez la page Origin Access Identity sur la CloudFront console ou utilisez-le ListCloudFrontOriginAccessIdentitiesdans l' CloudFront API.
- Remplacez *example-s3-bucket* par le nom de votre compartiment.

```
{
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::example-s3-bucket/*"
        }
    ]
}
```

Gestion des accès pour Amazon S3 Storage Lens

Accorder des autorisations pour Amazon S3 Storage Lens

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

S3 Storage Lens peut exporter vos métriques d'utilisation du stockage agrégées vers un compartiment Amazon S3 pour une analyse plus approfondie. Le compartiment dans lequel S3 Storage Lens place ses exportations de métriques porte le nom de compartiment de destination.

Lorsque vous configurez l'exportation des métriques S3 Storage Lens, vous devez disposer d'une politique de compartiment pour le compartiment de destination. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec Amazon S3 Storage Lens.

L'exemple de politique de compartiment suivant octroie à Amazon S3 l'autorisation d'écrire des objets (requêtes PUT) dans un compartiment de destination. Vous utilisez une stratégie de compartiment comme celle-ci sur le compartiment de destination lorsque vous configurez les mesures de S3 Storage Lens.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3StorageLensExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "storage-lens.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::destination-bucket/destination-prefix/
StorageLens/111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn:aws:s3:region-code:111122223333:storage-
lens/storage-lens-dashboard-configuration-id"
                }
            }
        }
    ]
}
```

Lorsque vous configurez une exportation de métriques au niveau de l'organisation S3 Storage Lens, utilisez la modification suivante de l'instruction `Resource` de la politique de compartiment précédente.

```
"Resource": "arn:aws:s3:::destination-bucket/destination-prefix/StorageLens/your-
organization-id/*",
```

Gestion des autorisations pour l'inventaire S3, les analyses S3 et les rapports d'inventaire S3

Accorder des autorisations pour l'inventaire S3 et les analyses S3.

L'inventaire S3 crée des listes d'objets dans un compartiment, et l'exportation de l'analyse de classe de stockage des analyses S3 crée des fichiers de sortie des données utilisées dans l'analyse. Le compartiment pour lequel l'inventaire répertorie les objets est appelé compartiment source. Le compartiment dans lequel le fichier d'inventaire ou le fichier d'exportation analytique est écrit est appelé compartiment de destination. Lorsque vous configurez un inventaire ou une exportation analytique, vous devez créer une politique de compartiment pour le compartiment de destination. Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3) et Analyses Amazon S3 - Analyse de classe de stockage.

L'exemple de politique de compartiment suivant accorde à Amazon S3 l'autorisation d'écrire des objets (requêtes PUT) à partir du compte pour le compartiment source vers le compartiment de destination. Vous utilisez une politique de compartiment comme celle-ci sur le compartiment de destination lors de la configuration de l'inventaire S3 et de l'exportation des analyses S3.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Sid": "InventoryAndAnalyticsExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
```

```
        }
    ]
}
```

Création de la configuration du rapport d'inventaire Control S3

Inventaire Simple Storage Service (Amazon S3) crée des listes des objets figurant dans un compartiment S3 et des métadonnées de chaque objet. L'`s3:PutInventoryConfiguration`autorisation permet à un utilisateur de créer une configuration d'inventaire qui inclut tous les champs de métadonnées d'objet disponibles par défaut et de spécifier le compartiment de destination pour stocker l'inventaire. Un utilisateur disposant d'un accès en lecture aux objets du compartiment de destination peut accéder à tous les champs de métadonnées d'objet disponibles dans ce rapport d'inventaire. Pour plus d'informations sur les champs de métadonnées disponibles dans S3 Inventory, consultez Liste d'inventaire Amazon S3.

Pour empêcher un utilisateur de configurer un rapport d'inventaire S3, supprimez l'`s3:PutInventoryConfiguration`autorisation de l'utilisateur.

Certains champs de métadonnées d'objet dans les configurations des rapports d'inventaire S3 sont facultatifs, ce qui signifie qu'ils sont disponibles par défaut mais qu'ils peuvent être restreints lorsque vous accordez l'`s3:PutInventoryConfiguration`autorisation à un utilisateur. Vous pouvez contrôler si les utilisateurs peuvent inclure ces champs de métadonnées facultatifs dans leurs rapports à l'aide de la clé de `s3:InventoryAccessibleOptionalFields` condition. Pour obtenir la liste des champs de métadonnées facultatifs disponibles dans S3 Inventory, consultez OptionalFieldsle manuel Amazon Simple Storage Service API Reference.

Pour autoriser un utilisateur à créer une configuration d'inventaire avec des champs de métadonnées facultatifs spécifiques, utilisez la clé de `s3:InventoryAccessibleOptionalFields` condition pour affiner les conditions de votre politique de compartiment.

L'exemple de politique suivant accorde à un utilisateur (*Ana*) l'autorisation de créer une configuration d'inventaire de manière conditionnelle. La `ForAllValues:StringEquals` condition de la politique utilise la clé de `s3:InventoryAccessibleOptionalFields` condition pour spécifier les deux champs de métadonnées facultatifs autorisés, à savoir `Size` et`StorageClass`. Ainsi, lors *Ana* de la création d'une configuration d'inventaire, les seuls champs de métadonnées facultatifs qu'elle peut inclure sont `Size` et`StorageClass`.

```
{
 "Id": "InventoryConfigPolicy",
```

```
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreationConditionally",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action":
    "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAllValues:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "Size",
        "StorageClass"
        ]
      }
    }
   }
 ]
}
```

Pour empêcher un utilisateur de configurer un rapport d'inventaire S3 qui inclut des champs de métadonnées facultatifs spécifiques, ajoutez une `Deny` déclaration explicite à la politique de compartiment pour le compartiment source. L'exemple de politique de compartiment suivant interdit à l'utilisateur *Ana* de créer une configuration d'inventaire dans le compartiment source *DOC-EXAMPLE-SOURCE-BUCKET* qui inclut les champs facultatifs `ObjectAccessControlList` ou de `ObjectOwner` métadonnées. L'utilisateur *Ana* peut toujours créer une configuration d'inventaire avec d'autres champs de métadonnées facultatifs.

```
{
 "Id": "InventoryConfigSomeFields",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreation",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
```

```
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",

  },
  {
   "Sid": "DenyCertainInventoryFieldCreation",
   "Effect": "Deny",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
     "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "ObjectOwner",
        "ObjectAccessControlList"
        ]
      }
    }
   }
 ]
}
```

> **Note**
>
> L'utilisation de la clé de `s3:InventoryAccessibleOptionalFields` condition dans les politiques relatives aux compartiments n'affecte pas la livraison des rapports d'inventaire basés sur les configurations d'inventaire existantes.

> **Important**
>
> Nous vous recommandons de l'utiliser `ForAllValues` avec un `Allow` effet ou `ForAnyValue` avec un `Deny` effet, comme indiqué dans les exemples précédents.
> Ne les utilisez pas `ForAllValues` avec `Deny` effet ni `ForAnyValue` avec `Allow` effet, car ces combinaisons peuvent être trop restrictives et bloquer la suppression de la configuration de l'inventaire.

Pour en savoir plus sur les opérateurs `ForAllValues` et les ensembles de `ForAnyValue` conditions, consultez la section Clés contextuelles à valeurs multiples du guide de l'utilisateur IAM.

Exigence d'une MFA

Amazon S3 prend en charge l'accès aux API protégé par MFA, une fonction qui peut appliquer l'authentification multi-facteur (MFA) pour l'accès à vos ressources Amazon S3. L'authentification multifactorielle fournit un niveau de sécurité supplémentaire que vous pouvez appliquer à votre AWS environnement. L'authentification MFA est une fonctionnalité de sécurité qui impose aux utilisateurs de prouver qu'ils détiennent physiquement un appareil d'authentification MFA en fournissant un code d'authentification MFA valide. Pour plus d'informations, consultez Authentification multifactorielle AWS. Vous pouvez exiger l'authentification MFA pour toutes les demandes d'accès à vos ressources Amazon S3.

Pour appliquer l'exigence d'authentification MFA, utilisez la clé de condition `aws:MultiFactorAuthAge` dans une politique de compartiment. Les utilisateurs IAM peuvent accéder aux ressources Amazon S3 à l'aide d'informations d'identification temporaires émises par le AWS Security Token Service (AWS STS). Vous fournissez le code MFA au moment de la demande AWS STS .

Quand Amazon S3 reçoit une demande d'authentification multifacteur, la clé de condition `aws:MultiFactorAuthAge` fournit une valeur numérique indiquant le temps écoulé (en secondes) depuis la création des informations d'identification temporaires. Si les informations d'identification temporaires fournies dans la demande n'ont pas été créées à l'aide d'un appareil d'authentification MFA, la valeur de cette clé est null (absente). Dans une politique de compartiment, vous pouvez ajouter une condition pour vérifier cette valeur, comme illustré dans l'exemple suivant.

Cet exemple de politique refuse toute opération Amazon S3 sur le dossier */taxdocuments* dans le compartiment *example-s3-bucket* si la demande n'a pas été authentifiée à l'aide de l'authentification MFA. Pour en savoir plus sur MFA, veuillez consulter la section Utilisation de l'authentification multifacteur (MFA) dans AWS du Guide de l'utilisateur IAM.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
```

```
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true }}
      }
    ]
 }
```

La condition `Null` figurant dans le bloc `Condition` équivaut à `true` si la valeur de la clé de condition `aws:MultiFactorAuthAge` est null, ce qui indique que les informations d'identification de sécurité temporaires figurant dans la demande ont été créées sans appareil d'authentification MFA.

La stratégie de compartiment suivante est une extension de la stratégie de compartiment précédente. La politique de compartiment suivante inclut deux déclarations de politique. Une déclaration accorde l'autorisation `s3:GetObject` sur un compartiment (*example-s3-bucket*) à tous. Une autre déclaration limite encore plus l'accès au dossier *example-s3-bucket/taxdocuments* du compartiment en requérant l'authentification MFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::example-s3-bucket/*"
      }
    ]
 }
```

Vous pouvez éventuellement utiliser une condition numérique pour limiter la durée de validité de la clé `aws:MultiFactorAuthAge`. La durée que vous spécifiez pour la clé `aws:MultiFactorAuthAge` est indépendante de la durée de vie de l'identifiant de sécurité temporaire utilisé pour authentifier la requête.

Par exemple, la stratégie de compartiment suivante, non seulement exige l'authentification MFA, mais vérifie également le temps écoulé depuis la création de la session temporaire. Cette stratégie refuse toute opération si la valeur de la clé `aws:MultiFactorAuthAge` indique que la session temporaire a été créée depuis plus d'une heure (3 600 secondes).

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
      },
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::example-s3-bucket/taxdocuments/*",
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
       },
       {
         "Sid": "",
         "Effect": "Allow",
         "Principal": "*",
         "Action": ["s3:GetObject"],
         "Resource": "arn:aws:s3:::example-s3-bucket/*"
       }
    ]
 }
```

Empêcher les utilisateurs de supprimer des objets

Par défaut, les utilisateurs ne disposent d'aucune autorisation. Mais lorsque vous créez des politiques, vous pouvez accorder aux utilisateurs des autorisations que vous n'aviez pas l'intention d'accorder. Pour éviter de telles failles d'autorisation, vous pouvez rédiger une politique d'accès plus stricte en ajoutant un refus explicite.

Pour empêcher explicitement les utilisateurs ou les comptes de supprimer des objets, vous devez ajouter les actions suivantes à une politique de compartiment : `s3:DeleteObjects`, `s3:DeleteObjectVersion`, et `s3:PutLifecycleConfiguration` autorisations. Les trois actions sont obligatoires car vous pouvez supprimer des objets soit en appelant explicitement l'API DELETE Object, soit en configurant leur cycle de vie (voir Gestion du cycle de vie de votre stockage) afin qu'Amazon S3 puisse supprimer les objets à l'expiration de leur durée de vie.

Dans l'exemple de politique suivant, vous refusez explicitement les autorisations DELETE Object à l'utilisateur Dave. Un refus explicite remplace toujours toute autre autorisation accordée.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::example-s3-bucket1",
   "arn:aws:s3:::example-s3-bucket1/*"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/Dave"
      },
```

```
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::example-s3-bucket1",
     "arn:aws:s3:::example-s3-bucket1/*"
      ]
    }
  ]
}
```

Exemples de politiques relatives aux compartiments utilisant des clés de condition

Vous pouvez utiliser le langage d'access policy pour spécifier des conditions lorsque vous accordez des autorisations. Vous pouvez utiliser l'élément `Condition` facultatif ou le bloc `Condition` pour spécifier des conditions lorsqu'une stratégie est appliquée.

Pour les stratégies qui utilisent des clés de condition Amazon S3 pour les opérations d'objet et de compartiment, consultez les exemples suivants. Pour plus d'informations sur les clés de condition, consultez Clés de conditions de politique pour Amazon S3. Pour obtenir la liste complète des actions, des clés de condition et des ressources Amazon S3 que vous pouvez spécifier dans les politiques, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Exemples - Clés de condition Amazon S3 pour les opérations sur les objets

Cette section fournit des exemples qui vous montrent comment utiliser des clés de condition spécifiques à Amazon S3 pour les opérations sur les objets. Pour obtenir la liste complète des actions, des clés de condition et des ressources Amazon S3 que vous pouvez spécifier dans les politiques, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Plusieurs exemples de stratégies montrent comment vous pouvez utiliser des clés de condition avec des opérations PUT Object. Les opérations PUT Object autorisent les entêtes spécifiques à la liste de contrôle d'accès (ACL) que vous pouvez utiliser pour accorder des autorisations basées sur les listes ACL. En utilisant ces clés, le propriétaire du compartiment peut définir une condition pour nécessiter des autorisations d'accès spécifiques quand l'utilisateur charge un objet. Vous pouvez également accorder des autorisations basées sur l'ACL lors de l'opération. PutObjectAcl Pour plus

d'informations, consultez PutObjectAcllе manuel Amazon S3 Amazon Simple Storage Service API Reference. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

Rubriques

- Exemple 1 : Octroi de s3 : PutObject autorisation nécessitant des objets stockés à l'aide du chiffrement côté serveur
- Exemple 2 : Octroi à s3 : PutObject autorisation de copier des objets avec une restriction sur la source de copie
- Exemple 3 : accorder l'accès à une version spécifique d'un objet
- Exemple 4 : Octroi d'autorisations en fonction des balises d'objets
- Exemple 5 : restriction de l'accès en fonction de l' Compte AWS ID du propriétaire du compartiment
- Exemple 6 : Exiger une version minimale de TLS

Exemple 1 : Octroi de s3 : PutObject autorisation nécessitant des objets stockés à l'aide du chiffrement côté serveur

Supposons que le Compte A possède un compartiment. L'administrateur du compte souhaite accorder à Jane, une utilisatrice du Compte A, l'autorisation de télécharger des objets avec comme condition que Jane demande toujours le chiffrement côté serveur afin qu'Amazon S3 enregistre les objets chiffrés. L'administrateur du compte A peut accomplir cela en utilisant la clé de condition `s3:x-amz-server-side-encryption` comme illustré. La paire de clé-valeur dans le bloc `Condition` spécifie la clé `s3:x-amz-server-side-encryption`.

```
"Condition": {
     "StringNotEquals": {
         "s3:x-amz-server-side-encryption": "AES256"
     }}
```

Lorsque vous testez l'autorisation à l'aide du AWS CLI, vous devez ajouter le paramètre requis à l'aide du `--server-side-encryption` paramètre.

```
aws s3api put-object --bucket example1bucket --key HappyFace.jpg --body c:
\HappyFace.jpg --server-side-encryption "AES256" --profile AccountBadmin
```

Exemple 2 : Octroi à s3 : PutObject autorisation de copier des objets avec une restriction sur la source de copie

Dans la demande PUT Object, quand vous spécifiez un objet source, il s'agit d'une opération de copie (veuillez consulter PUT Object - Copy). En conséquence, le propriétaire du compartiment peut octroyer une autorisation utilisateur pour copier des objets avec des restrictions sur la source, par exemple :

- Autoriser la copie d'objets uniquement à partir du compartiment `sourcebucket`.
- Autoriser la copie d'objets à partir du compartiment source, et uniquement les objets dont le préfixe de nom de clé commence par public/ f (par exemple, sourcebucket/public/*).
- Autoriser la copie uniquement d'un objet spécifique à partir du compartiment source (par exemple, sourcebucket/example.jpg).

La stratégie de compartiment suivante accorde à l'utilisateur (Dave) l'autorisation `s3:PutObject`. Celle-ci lui permet de copier des objets uniquement à la condition que la demande inclue l'en-tête `s3:x-amz-copy-source` et que la valeur de l'en-tête spécifie le préfixe de nom de clé `/awsexamplebucket1/public/*`.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Sid": "cross-account permission to user in your own account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*"
        },
        {
            "Sid": "Deny your user permission to upload object if copy source is not /
bucket/folder",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
```

```
            "Condition": {
                "StringNotLike": {
                    "s3:x-amz-copy-source": "awsexamplebucket1/public/*"
                }
            }
        }
    ]
}
```

Testez la politique à l'aide du AWS CLI

Vous pouvez tester l'autorisation à l'aide de la AWS CLI `copy-object` commande. Vous spécifiez la source en ajoutant le paramètre `--copy-source`, le préfixe du nom de clé doit correspondre au préfixe autorisé dans la stratégie. Vous devez fournir à l'utilisateur Dave les informations d'identification en utilisant le paramètre `--profile`. Pour plus d'informations sur la configuration du AWS CLI, consultezDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

```
aws s3api copy-object --bucket awsexamplebucket1 --key HappyFace.jpg
--copy-source examplebucket/public/PublicHappyFace1.jpg --profile AccountADave
```

Octroyer une autorisation pour copier uniquement un objet spécifique

La stratégie précédente utilise la condition `StringNotLike`. Pour octroyer l'autorisation de copier uniquement un objet spécifique, vous devez changer la condition de `StringNotLike` à `StringNotEquals`, puis spécifier la clé d'objet exacte comme indiqué.

```
"Condition": {
       "StringNotEquals": {
           "s3:x-amz-copy-source": "awsexamplebucket1/public/PublicHappyFace1.jpg"
       }
}
```

Exemple 3 : accorder l'accès à une version spécifique d'un objet

Supposons que le Compte A possède un compartiment activé pour la version. Le compartiment a plusieurs versions de l'objet `HappyFace.jpg`. L'administrateur de compte souhaite maintenant octroyer à son utilisateur Dave l'autorisation d'obtenir uniquement une version spécifique de l'objet. L'administrateur de compte peut accomplir cela en octroyant à Dave l'autorisation `s3:GetObjectVersion` de manière conditionnelle comme indiqué. La paire de clé-valeur dans le bloc `Condition` spécifie la clé de condition `s3:VersionId`. Dans ce cas, Dave doit connaître l'ID exact de la version de l'objet pour récupérer l'objet.

Pour plus d'informations, consultez GetObjectle manuel Amazon Simple Storage Service API Reference.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg"
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::examplebucketversionenabled/HappyFace.jpg",
            "Condition": {
                "StringNotEquals": {
                    "s3:VersionId": "AaaHbAQitwiL_h47_44lRO2DDfLlBO5e"
                }
            }
        }
    ]
}
```

Testez la politique à l'aide du AWS CLI

Vous pouvez tester les autorisations à l'aide de la AWS CLI `get-object` commande avec le `--version-id` paramètre identifiant la version spécifique de l'objet. La commande récupère l'objet et l'enregistre dans le fichier `OutputFile.jpg`.

```
aws s3api get-object --bucket examplebucketversionenabled --key HappyFace.jpg
 OutputFile.jpg --version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

Exemple 4 : Octroi d'autorisations en fonction des balises d'objets

Pour obtenir des exemples sur l'utilisation des clés de condition de balisage d'objet avec des opérations Amazon S3, veuillez consulter Stratégies de balisage et de contrôle d'accès.

Exemple 5 : restriction de l'accès en fonction de l' Compte AWS ID du propriétaire du compartiment

Vous pouvez utiliser la clé `aws:ResourceAccount` ou `s3:ResourceAccount` pour écrire des politiques IAM ou des politiques de point de terminaison du cloud privé virtuel (VPC) qui restreignent l'accès des utilisateurs, des rôles ou des applications aux compartiments Amazon S3 appartenant à un ID de Compte AWS spécifique. Vous pouvez utiliser cette clé de condition si vous souhaitez empêcher les clients de votre VPC d'accéder aux compartiments dont vous n'êtes pas le propriétaire.

Sachez toutefois que certains AWS services dépendent de l'accès à des buckets AWS gérés. Par conséquent, l'utilisation de la clé `aws:ResourceAccount` ou `s3:ResourceAccount` dans votre politique IAM risque également d'avoir un impact sur l'accès à ces ressources.

Pour plus d'informations et d'exemples, veuillez consulter les ressources suivantes :

- Restreindre l'accès aux compartiments dans un Compte AWS spécifié dans le Guide AWS PrivateLink
- Restreindre l'accès aux compartiments utilisés par Amazon ECR dans le Guide de l'utilisateur Amazon ECR
- Fournissez l'accès requis à Systems Manager pour les buckets Amazon S3 AWS gérés dans le Guide AWS Systems Manager
- Limit access to Amazon S3 buckets owned by specific Comptes AWS (Limiter l'accès aux compartiments Amazon S3 détenus par des comptes AWS spécifiques) sur la page AWS Storage Blog

Exemple 6 : Exiger une version minimale de TLS

Vous pouvez utiliser la clé de TlsVersion condition s3 : pour écrire des politiques IAM, Virtual Private Cloud Endpoint (VPCE) ou de bucket qui limitent l'accès des utilisateurs ou des applications aux compartiments Amazon S3 en fonction de la version TLS utilisée par le client. Vous pouvez utiliser cette clé de condition pour écrire des stratégies qui nécessitent une version TLS minimale.

Example

Cet exemple de politique de compartiment refuse les PutObject demandes des clients dont la version TLS est inférieure à 1.2, par exemple 1.1 ou 1.0.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::example-s3-bucket1",
                "arn:aws:s3:::example-s3-bucket1/*"
            ],
            "Condition": {
                "NumericLessThan": {
                    "s3:TlsVersion": 1.2
                }
            }
        }
    ]
}
```

Example

Cet exemple de politique de compartiment autorise les PutObject demandes des clients dont la version TLS est supérieure à 1.1, par exemple 1.2, 1.3 ou supérieure.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::example-s3-bucket1",
                "arn:aws:s3:::example-s3-bucket1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
```

```
            }
        }
    ]
}
```

Exemples - Clés de condition Amazon S3 pour les opérations sur les compartiments

Cette section fournit des exemples de stratégies qui vous montrent comment utiliser des clés de condition spécifiques à Amazon S3 pour les opérations sur les compartiments.

Rubriques

- Exemple 1 : Octroi d'une GetObject autorisation s3 : assortie d'une condition sur une adresse IP
- Exemple 2 : Obtention d'une liste d'objets dans un compartiment avec un préfixe spécifique
- Exemple 3 : Définition du nombre maximal de clés

Exemple 1 : Octroi d'une GetObject autorisation s3 : assortie d'une condition sur une adresse IP

Vous pouvez autoriser les utilisateurs authentifiés à utiliser l'`s3:GetObject`action si la demande provient d'une plage d'adresses IP spécifique (192.0.2.*), sauf si l'adresse IP est 192.0.2.188. Dans le bloc de condition, les éléments `IpAddress` et `NotIpAddress` sont des conditions, chacune disposant d'une paire clé-valeur pour évaluation. Dans cet exemple, les deux paires clé-valeur utilisent la clé `aws:SourceIp` AWS-wide.

> **Note**
>
> Les valeurs de clé `IPAddress` et `NotIpAddress` spécifiées dans la condition utilisent la notation CIDR comme décrit dans RFC 4632. Pour plus d'informations, consultez http://www.rfc-editor.org/rfc/rfc4632.txt.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": "*",
```

```
            "Action":"s3:GetObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition" : {
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "192.0.2.188/32"
                }
            }
        }
    ]
}
```

Vous pouvez également utiliser d'autres AWS clés de condition générales dans les politiques Amazon S3. Par exemple, vous pouvez spécifier les clés de condition `aws:SourceVpce` et `aws:SourceVpc` dans les stratégies de compartiment pour les points de terminaison de VPC. Pour plus d'exemples, consultez Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.

> **Note**
>
> Pour certaines clés de condition AWS globales, seuls certains types de ressources sont pris en charge. Par conséquent, vérifiez si Amazon S3 prend en charge la clé de condition globale et le type de ressource que vous souhaitez utiliser, ou si vous devez utiliser une clé de condition spécifique à Amazon S3 à la place. Pour obtenir la liste complète des types de ressources et des clés de condition pris en charge pour Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Exemple 2 : Obtention d'une liste d'objets dans un compartiment avec un préfixe spécifique

Vous pouvez utiliser la clé de `s3:prefix` condition pour limiter la réponse de l'API GET Bucket (ListObjects) aux noms de clés dotés d'un préfixe spécifique. Si vous êtes le propriétaire du compartiment, vous pouvez limiter un utilisateur pour répertorier le contenu d'un préfixe spécifique dans le compartiment. Cette clé de condition est utile si les objets du compartiment sont organisés par préfixe de nom de clé. La console Amazon S3 utilise des préfixes de nom de clé pour afficher un concept de dossier. Seule la console prend en charge le concept de dossiers ; l'API Amazon S3 ne prend en charge que les compartiments et les objets. Pour plus d'informations sur l'utilisation de

préfixes et de délimiteurs pour filtrer les autorisations d'accès, consultez Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur.

Par exemple, si vous avez deux objets avec les noms de clés `public/object1.jpg` et `public/object2.jpg`, la console affiche les objets dans le dossier `public`. Dans l'API Amazon S3, il s'agit d'objets avec des préfixes, pas d'objets dans des dossiers. Cependant, dans l'API Amazon S3, si vous organisez vos clés d'objet en utilisant de tels préfixes, vous pouvez octroyer l'autorisation `s3:ListBucket` avec la condition `s3:prefix`, ce qui permet à l'utilisateur d'obtenir une liste des noms de clés avec ces préfixes spécifiques.

Dans cet exemple, le propriétaire du compartiment et le compte parent auquel appartient l'utilisateur sont identiques. Ainsi, le propriétaire du compartiment peut utiliser une stratégie de compartiment ou une stratégie d'utilisateur. Pour plus d'informations sur les autres clés de condition que vous pouvez utiliser avec l'API GET Bucket (ListObjects), consultez ListObjects.

Stratégie utilisateur

La stratégie d'utilisateur suivante octroie l'autorisation `s3:ListBucket` (veuillez consulter GET Bucket (List Objects)) avec une condition qui nécessite que l'utilisateur spécifie le préfixe `prefix` dans la demande avec la valeur `projects`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:ListBucket",
         "Resource":"arn:aws:s3:::awsexamplebucket1",
         "Condition" : {
             "StringEquals" : {
                 "s3:prefix": "projects"
             }
          }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Action": "s3:ListBucket",
         "Resource": "arn:aws:s3:::awsexamplebucket1",
         "Condition" : {
             "StringNotEquals" : {
```

```
                "s3:prefix": "projects"
            }
         }
      }
   ]
}
```

La condition restreint l'utilisateur à répertorier les clés d'objets avec le préfixe `projects`. Le refus explicite ajouté refuse la demande de l'utilisateur de répertorier les clés avec un autre préfixe, peu importe les autres autorisations dont dispose l'utilisateur. Par exemple, il est possible que l'utilisateur obtienne l'autorisation de répertorier les clés d'objets sans aucune restriction, grâce aux mises à jour de la stratégie d'utilisateur précédente ou via une stratégie de compartiment. Dans la mesure où le refus explicite a toujours priorité, la demande de l'utilisateur de répertorier des clés autres que le préfixe `projects` est refusée.

Stratégie de compartiment

Si vous ajoutez l'élément `Principal` à la stratégie utilisateur ci-dessus, identifiant l'utilisateur, vous avez maintenant une stratégie de compartiment comme indiqué.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
         },
         "Action":  "s3:ListBucket",
         "Resource": "arn:aws:s3:::awsexamplebucket1",
         "Condition" : {
             "StringEquals" : {
                 "s3:prefix": "projects"
             }
          }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
```

```
            },
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::awsexamplebucket1",
            "Condition" : {
                "StringNotEquals" : {
                    "s3:prefix": "projects"
                }
             }
        }
    ]
}
```

Testez la politique à l'aide du AWS CLI

Vous pouvez tester la politique à l'aide de la `list-object` AWS CLI commande suivante. Dans la commande, vous fournissez les informations d'identification utilisateur en utilisant le paramètre `--profile`. Pour plus d'informations sur la configuration et l'utilisation du AWS CLI, consultezDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

```
aws s3api list-objects --bucket awsexamplebucket1 --prefix examplefolder --profile
 AccountADave
```

Si le compartiment est activé pour la version, pour répertorier les objets dans le compartiment, vous devez accorder l'autorisation `s3:ListBucketVersions` dans la stratégie précédente, au lieu de l'autorisation `s3:ListBucket`. Cette autorisation prend également en charge la clé de condition `s3:prefix`.

Exemple 3 : Définition du nombre maximal de clés

Vous pouvez utiliser la clé de `s3:max-keys` condition pour définir le nombre maximum de clés que le demandeur peut renvoyer dans un bucket (ListObjects) ou une ListObjectVersionsdemande GET. Par défaut, l'API retourne jusqu'à 1 000 clés. Pour obtenir la liste des opérateurs de conditions numériques que vous pouvez utiliser avec `s3:max-keys` et les exemples qui les accompagnent, veuillez consulter Opérateurs de conditions numériques dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon S3

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources Amazon S3. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs

des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par Amazon S3, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference.

Rubriques

- Bonnes pratiques en matière de politiques
- Exemples de politiques basées sur l'identité pour Amazon S3
- Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources Amazon S3 dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.

- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour Amazon S3

Cette section présente plusieurs exemples de politiques basées sur l'identité AWS Identity and Access Management (IAM) pour contrôler l'accès à Amazon S3. Par exemple, les politiques de compartiment (politiques basées sur les ressources), voir. Politiques relatives aux compartiments pour Amazon S3 Pour en savoir plus sur le langage des politiques IAM, consultez Politiques et autorisations dans Amazon S3.

Les exemples de politique suivants fonctionnent si vous les testez par programmation. Cependant, pour les utiliser avec la console Amazon S3, vous devez accorder des autorisations supplémentaires qui sont requises par la console. Pour des informations sur l'utilisation de stratégies comme celles-ci avec la console Amazon S3, veuillez consulter Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur.

Rubriques

Autoriser un accès utilisateur IAM à l'un de vos compartiments

Dans cet exemple, vous souhaitez autoriser un utilisateur IAM à Compte AWS accéder à l'un de vos compartiments, *`example-s3-bucket1`*, et lui permettre d'ajouter, de mettre à jour et de supprimer des objets.

En plus de l'octroi des autorisations `s3:PutObject`, `s3:GetObject` et `s3:DeleteObject` à l'utilisateur, la stratégie octroie aussi les autorisations `s3:ListAllMyBuckets`, `s3:GetBucketLocation` et `s3:ListBucket`. Ces conditions supplémentaires sont requises par la console. De la même manière, les actions `s3:PutObjectAcl` et `s3:GetObjectAcl` sont nécessaires pour que les objets puissent être copiés, coupés et collés dans la console. Pour afficher un exemple qui octroie les autorisations aux utilisateurs et qui les teste en utilisant la console, consultez Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": "s3:ListAllMyBuckets",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
```

```
         "Action":["s3:ListBucket","s3:GetBucketLocation"],
         "Resource":"arn:aws:s3:::example-s3-bucket1"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/*"
      }
   ]
}
```

Autoriser chaque utilisateur IAM à accéder à un dossier dans un compartiment

Dans cet exemple, vous souhaitez que deux utilisateurs IAM, Mary et Carlos, aient accès à votre bucket, *example-s3-bucket1*, afin qu'ils puissent ajouter, mettre à jour et supprimer des objets. Cependant, vous souhaitez restreindre l'accès de chaque utilisateur à un seul préfixe (dossier) dans le compartiment. Vous pouvez créer des dossiers dont le nom correspond à leur nom d'utilisateur.

```
example-s3-bucket1
   Mary/
   Carlos/
```

Pour uniquement octroyer à chaque utilisateur l'accès à son dossier, vous pouvez écrire une stratégie pour chaque utilisateur et l'attacher individuellement. Par exemple, vous pouvez attacher la politique suivante à l'utilisateur Mary pour lui octroyer des autorisations spécifiques à Amazon S3 sur le dossier *example-s3-bucket1*/*Mary*.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
```

```
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/Mary/*"
      }
   ]
}
```

Vous pouvez alors attacher une stratégie similaire à l'utilisateur Carlos, en spécifiant le dossier *Carlos* dans la valeur `Resource`.

Au lieu d'attacher des politiques à des utilisateurs individuels, vous pouvez en écrire une seule qui utilise une variable de politique, puis attacher la politique à un groupe. Vous devez d'abord créer un groupe et ajouter Mary et Carlos à ce groupe. L'exemple de stratégie suivant octroie un ensemble d'autorisations Amazon S3 dans le dossier *example-s3-bucket1*`/${aws:username}`. Lorsque la politique est évaluée, la variable de stratégie `${aws:username}` est remplacée par le nom d'utilisateur du demandeur. Par exemple, si Mary envoie une demande pour placer un objet, l'opération est autorisée uniquement si Mary télécharge l'objet dans le dossier *example-s3-bucket1*`/Mary`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/${aws:username}/*"
      }
   ]
}
```

> **Note**
>
> Lorsque vous utilisez des variables de politique, vous devez spécifier explicitement la version `2012-10-17` de la politique. La version par défaut du langage de politique IAM, 2008-10-17, ne prend pas en charge les variables de stratégie.

Si vous souhaitez tester la politique précédente sur la console Amazon S3, la console nécessite des autorisations supplémentaires, comme indiqué dans la politique suivante. Pour en savoir plus sur la façon dont la console utilise ces autorisations, consultez Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur.

```
{
 "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
       "s3:ListAllMyBuckets",
       "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example-s3-bucket1",
      "Condition":{
            "StringEquals":{
                   "s3:prefix":[""], "s3:delimiter":["/"]
                         }
               }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example-s3-bucket1",
      "Condition":{  "StringLike":{"s3:prefix":["${aws:username}/*"] }
       }
```

```
    },
      {
     "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/${aws:username}/*"
      }
  ]
}
```

> **Note**
>
> Dans la version 2012-10-17 de la stratégie, les variables de stratégie commencent par $. Ce changement de syntaxe peut potentiellement créer un conflit si votre clé d'objet (nom d'objet) inclut un $.
> Pour éviter ce conflit, spécifiez le caractère $ en utilisant `${$}`. Par exemple, pour inclure la clé d'objet `my$file` dans une politique, spécifiez `my${$}file`.

Bien que les noms d'utilisateur IAM soient des identifiants conviviaux et compréhensibles, il n'est pas nécessaire qu'ils soient uniques à l'échelle mondiale. Par exemple, si l'utilisateur Carlos quitte l'organisation et qu'un autre Carlos la rejoint, le nouvel employé Carlos pourrait accéder aux informations de l'ancien employé Carlos.

Au lieu d'utiliser des noms d'utilisateur, vous pouvez créer des dossiers basés sur les identifiants utilisateur IAM. Chaque ID d'utilisateur IAM est unique. Dans ce cas, vous devez modifier la stratégie précédente pour utiliser la variable de stratégie `${aws:userid}`. Pour plus d'informations sur les identifiants utilisateur, consultez Identificateurs IAM dans le Guide de l'utilisateur IAM.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
```

```
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/home/${aws:userid}/*"
      }
   ]
}
```

Autoriser des utilisateurs non IAM (utilisateurs d'application mobile) à accéder à des dossiers dans un compartiment

Supposons que vous souhaitez développer une application mobile, un jeu qui stocke les données utilisateur dans un compartiment S3. Pour chaque utilisateur d'application, vous souhaitez créer un dossier dans votre compartiment. Vous souhaitez également limiter l'accès de chaque utilisateur à son propre dossier. Mais vous ne pouvez pas créer de dossiers avant que quelqu'un ne télécharge votre application et ne commence à jouer au jeu, car vous n'avez pas son nom d'utilisateur.

Dans ce cas, vous pouvez demander que les utilisateurs se connectent à votre application en utilisant des fournisseurs d'identité publics tels que Login with Amazon, Facebook ou Google. Une fois que les utilisateurs se sont connectés à votre application via l'un de ces fournisseurs, ils ont un ID utilisateur que vous pouvez utiliser afin de créer des dossiers spécifiques aux utilisateurs au moment de l'exécution.

Vous pouvez ensuite utiliser la fédération d'identité Web AWS Security Token Service pour intégrer les informations du fournisseur d'identité à votre application et obtenir des informations de sécurité temporaires pour chaque utilisateur. Vous pouvez ensuite créer des stratégies IAM qui permettent à l'application d'accéder à votre compartiment et d'effectuer des opérations comme la création de dossiers spécifiques à l'utilisateur et le chargement de données. Pour plus d'informations sur la fédération d'identités Web, consultez À propos de la fédération d'identité Web dans le Guide de l'utilisateur IAM.

Autoriser un groupe à avoir un dossier partagé dans Amazon S3

Le fait d'attacher la stratégie suivante au groupe octroie à tous les membres du groupe l'accès au dossier suivant dans Amazon S3 : *example-s3-bucket1*`/share/marketing`. Les membres du groupe sont uniquement autorisés à accéder aux autorisations spécifiques à Amazon S3 indiquées dans la stratégie et uniquement pour les objets dans le dossier spécifié.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/share/marketing/*"
      }
   ]
}
```

Autoriser tous vos utilisateurs à lire des objets dans une partie d'un compartiment

Dans cet exemple, vous créez un groupe appelé *AllUsers*, qui contient tous les utilisateurs IAM que possède le Compte AWS. Vous attachez ensuite une stratégie qui donne au groupe l'accès à `GetObject` et à `GetObjectVersion`, mais uniquement pour les objets dans le dossier *example-s3-bucket1*`/readonly`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket1/readonly/*"
      }
   ]
}
```

Autoriser un partenaire à déposer des fichiers dans une partie spécifique d'un compartiment

Dans cet exemple, vous créez un groupe appelé *AnyCompany* qui représente une société partenaire. Vous créez un utilisateur IAM pour la personne ou l'application spécifique de la société partenaire ayant besoin d'un accès, puis vous placez l'utilisateur dans le groupe.

Vous attachez ensuite une politique qui donne au groupe l'accès `PutObject` au dossier suivant dans le compartiment :

example-s3-bucket1/uploads/anycompany

Vous voulez empêcher le groupe *AnyCompany* de faire quoi que soit d'autre avec le compartiment et vous ajoutez donc une instruction qui refuse explicitement l'autorisation à toutes les actions Amazon S3 à l'exception de `PutObject` sur n'importe quelle ressource Amazon S3 du Compte AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::example-s3-bucket1/uploads/anycompany/*"
      },
      {
         "Effect":"Deny",
         "Action":"s3:*",
         "NotResource":"arn:aws:s3:::example-s3-bucket1/uploads/anycompany/*"
      }
   ]
}
```

Restriction de l'accès aux compartiments Amazon S3 dans un Compte AWS spécifique

Si vous voulez vous assurer que vos principaux Amazon S3 accèdent uniquement aux ressources qui se trouvent dans un environnement sécurisé Compte AWS, vous pouvez restreindre l'accès. Par exemple, cette politique IAM basée sur une identité utilise un effet `Deny` pour bloquer l'accès aux actions Amazon S3, sauf si la ressource Amazon S3 concernée appartient au compte *222222222222*. Pour empêcher le principal IAM Compte AWS d'accéder aux objets Amazon S3 en dehors du compte, joignez la politique IAM suivante :

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
    }
  ]
}
```

> **Note**
>
> Cette politique ne remplace pas vos contrôles d'accès IAM existants, car elle n'accorde aucun accès. Cette politique agit plutôt comme une barrière de protection supplémentaire pour vos autres autorisations IAM, quelles que soient les autorisations accordées par d'autres politiques IAM.

Assurez-vous de remplacer l'ID de compte *222222222222* dans la politique par votre propre Compte AWS. Pour appliquer une politique à plusieurs comptes tout en conservant cette restriction, remplacez l'ID de compte par la clé de condition `aws:PrincipalAccount`. Cette condition exige que le principal et la ressource soient dans le même compte.

Restreindre l'accès aux compartiments Amazon S3 au sein de votre unité organisationnelle

Si vous avez configuré une unité organisationnelle (UO) AWS Organizations, vous souhaiterez peut-être restreindre l'accès au compartiment Amazon S3 à une partie spécifique de votre organisation. Dans cet exemple, nous utiliserons la clé `aws:ResourceOrgPaths` pour restreindre l'accès au compartiment Amazon S3 à une unité organisationnelle de votre organisation. Dans cet exemple, l'ID de l'unité organisationnelle est *ou-acroot-exampleou*. Assurez-vous de remplacer cette valeur dans votre politique par vos propres ID d'unité organisationnelle.

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "AllowS3AccessOutsideMyBoundary",
       "Effect": "Allow",
       "Action": [
         "s3:*"
       ],
       "Resource": "*",
       "Condition": {
         "ForAllValues:StringNotLike": {
           "aws:ResourceOrgPaths": [
             "o-acorg/r-acroot/ou-acroot-exampleou/"
           ]
         }
       }
     }
   ]
 }
```

> **Note**
>
> Cette politique n'accorde aucun accès. Au lieu de cela, cette politique sert de backstop pour vos autres autorisations IAM, empêchant vos principaux d'accéder aux objets Amazon S3 en dehors d'une limite définie par l'unité organisationnelle.

La politique refuse l'accès aux actions Amazon S3, sauf si l'objet Amazon S3 auquel on accède se trouve dans l'unité organisationnelle *ou-acroot-exampleou* de votre organisation. La condition de politique IAM a besoin de `aws:ResourceOrgPaths`, clé de condition à valeurs multiples, pour contenir un des chemins d'unité organisationnelle répertoriés. La politique utilise l'opérateur `ForAllValues:StringNotLike` pour comparer les valeurs de `aws:ResourceOrgPaths` aux unités organisationnelles répertoriées sans correspondance avec respect de la casse.

Restriction de l'accès aux compartiments Amazon S3 au sein de votre organisation

Pour restreindre l'accès aux objets Amazon S3 au sein de votre organisation, attachez une politique IAM à la racine de l'organisation, en l'appliquant à tous les comptes de votre organisation. Pour demander à vos principaux IAM de respecter cette règle, utilisez une politique de contrôle des

services (SCP). Si vous choisissez d'utiliser une SCP, veillez à bien tester la SCP avant d'attacher la politique à la racine de l'organisation.

Dans l'exemple de politique suivant, l'accès est refusé aux actions Amazon S3, sauf si l'objet Amazon S3 accédé se trouve dans la même organisation que le principal IAM qui y accède :

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "DenyS3AccessOutsideMyBoundary",
       "Effect": "Deny",
       "Action": [
         "s3:*"
       ],
       "Resource": "arn:aws:s3:::*/*",
       "Condition": {
         "StringNotEquals": {
           "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
         }
       }
     }
   ]
 }
```

> **Note**
>
> Cette politique n'accorde aucun accès. Cette politique agit plutôt comme un backstop pour vos autres autorisations IAM, empêchant vos principaux d'accéder à tous les objets Amazon S3 en dehors de votre organisation. Cette politique s'applique également aux ressources Amazon S3 créées après l'entrée en vigueur de la politique.

Dans cet exemple, la condition de politique IAM nécessite que `aws:ResourceOrgID` et `aws:PrincipalOrgID` soient égaux les uns aux autres. Avec cette exigence, le principal qui fait la demande et la ressource accédée doivent faire partie de la même organisation.

Octroi de l'autorisation de récupérer la PublicAccessBlock configuration d'un Compte AWS

L'exemple de politique basée sur l'identité suivant accorde l'`s3:GetAccountPublicAccessBlock`autorisation à un utilisateur. Pour ces autorisations,

définissez la valeur `Resource` sur `"*"`. Pour plus d'informations sur les ARN des ressources, consultezRessources relatives aux politiques pour Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Restreindre la création de compartiments à une seule région

Supposons qu'un Compte AWS administrateur souhaite accorder à son utilisateur (Dave) l'autorisation de créer un bucket dans la région Amérique du Sud (São Paulo) uniquement. L'administrateur de compte peut attacher la stratégie d'utilisateur suivant octroyant l'autorisation `s3:CreateBucket` avec une condition comme indiqué. La paire de clé-valeur dans le bloc `Condition` spécifie la clé `s3:LocationConstraint` et la Région `sa-east-1` comme sa valeur.

> **Note**
>
> Dans cet exemple, le propriétaire du compartiment octroie l'autorisation à un de ses utilisateurs. Par conséquent une stratégie de compartiment ou une stratégie d'utilisateur peuvent être utilisées. Cette exemple illustre une stratégie d'utilisateur.

Pour obtenir la liste des régions Amazon S3, consultez Régions et points de terminaison dans Références générales AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
```

```
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       }
    ]
}
```

Ajouter un refus explicite

La stratégie précédente limite la création, par l'utilisateur, d'un compartiment dans une autre Région à l'exception de `sa-east-1`. Toutefois, une autre stratégie peut accorder à cet utilisateur l'autorisation de créer des compartiments dans une autre Région. Par exemple, si l'utilisateur appartient à un groupe, une stratégie peut être attachée à ce groupe, autorisant tous les utilisateurs du groupe à créer des compartiments dans une autre Région. Pour garantir que l'utilisateur n'est pas autorisé à créer des buckets dans une autre région, vous pouvez ajouter une déclaration de refus explicite dans la politique ci-dessus.

L'instruction `Deny` utilise la condition `StringNotLike`. C'est-à-dire qu'une demande de création de compartiment est refusée si la contrainte d'emplacement n'est pas `sa-east-1`. Le refus explicite n'autorise pas l'utilisateur à créer un bucket dans une autre région, quelle que soit l'autre autorisation qu'il obtient. La politique suivante inclut une déclaration de refus explicite.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
```

```
        },
       {
          "Sid":"statement2",
          "Effect":"Deny",
          "Action": "s3:CreateBucket",
          "Resource": "arn:aws:s3:::*",
          "Condition": {
              "StringNotLike": {
                  "s3:LocationConstraint": "sa-east-1"
              }
          }
        }
    ]
}
```

Testez la politique à l'aide du AWS CLI

Vous pouvez tester la politique à l'aide de la `create-bucket` AWS CLI commande suivante. Cet exemple utilise le fichier `bucketconfig.txt` pour spécifier la contrainte d'emplacement. Notez le chemin Windows du fichier. Vous devez mettre à jour le nom du compartiment et le chemin comme approprié. Vous devez fournir les informations d'identification utilisateur en utilisant le paramètre `--profile`. Pour plus d'informations sur la configuration et l'utilisation du AWS CLI, consultezDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-
configuration file://c:/Users/someUser/bucketconfig.txt
```

Le fichier `bucketconfig.txt` spécifie la configuration comme suit.

```
{"LocationConstraint": "sa-east-1"}
```

Contrôle de l'accès à un compartiment avec des stratégies d'utilisateur

Cette démonstration explique comment les autorisations utilisateur fonctionnent avec Amazon S3. Dans cet exemple, vous créez un compartiment avec des dossiers. Vous créez ensuite des utilisateurs AWS Identity and Access Management IAM dans votre compartiment Amazon S3 Compte AWS et les dossiers qu'il contient et leur accordez des autorisations supplémentaires.

Rubriques

- Principes de base des compartiments et des dossiers

Principes de base des compartiments et des dossiers

Le modèle de données Amazon S3 est une structure horizontale : vous créez un compartiment et ce compartiment stocke des objets. Il n'existe aucune hiérarchie de sous-compartiments ou de sous-dossiers, mais vous pouvez émuler une hiérarchie de dossiers. Des outils tels que la console Amazon S3 peuvent présenter une vue de ces dossiers et sous-dossiers logiques dans votre compartiment.

La console montre qu'un compartiment nommé `companybucket` comporte trois dossiers, `Private`, `Development` et `Finance`, ainsi qu'un objet, `s3-dg.pdf`. La console utilise les noms d'objet (clés) pour créer une hiérarchie logique avec des dossiers et des sous-dossiers. Considérez les exemples suivants :

- Lorsque vous créez le dossier `Development`, la console crée un objet avec la clé `Development/`. Notez la barre oblique de fin (`/`) comme délimiteur.
- Lorsque vous chargez un objet nommé `Projects1.xls` dans le dossier `Development`, la console charge l'objet et lui fournit la clé `Development/Projects1.xls`.

  Dans la clé, `Development` est le préfixe et `/` le délimiteur. L'API Amazon S3 prend en charge les préfixes et les délimiteurs dans ses opérations. Par exemple, vous pouvez obtenir la liste de tous les objets d'un compartiment avec un préfixe et un délimiteur spécifiques. Dans la console, quand ouvrez le dossier `Development`, la console répertorie les objets de ce dossier. Dans l'exemple suivant, le dossier `Development` contient un seul objet.

Lorsque la console répertorie le dossier `Development` dans le compartiment `companybucket`, elle envoie une demande à Amazon S3 dans laquelle elle spécifie le préfixe `Development` et le délimiteur `/`. La réponse de la console ressemble à une liste de dossiers du système de fichiers de votre ordinateur. L'exemple précédent montre que le compartiment `companybucket` possède un objet doté de la clé `Development/Projects1.xls`.

La console utilise des clés d'objet pour déduire une hiérarchie logique. Amazon S3 ne possède aucune hiérarchie physique. Amazon S3 ne propose que des compartiments contenant des objets dans une structure de fichier plate. Lorsque vous créez des objets à l'aide de l'API Amazon S3, vous pouvez utiliser des clés d'objet qui impliquent une hiérarchie logique. Lorsque vous créez une hiérarchie logique d'objets, vous pouvez gérer l'accès aux dossiers individuels, comme le montre cette procédure détaillée.

Avant de commencer, veillez à vous familiariser avec le concept de contenu du compartiment de niveau racine. Supposez que votre compartiment `companybucket` contienne les objets suivants :

- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

Ces clés d'objet créent une hiérarchie logique avec `Private`, `Development` et `Finance` comme dossiers de niveau racine et `s3-dg.pdf` comme objet de niveau racine. Lorsque vous choisissez le nom du compartiment dans la console Amazon S3, les éléments de niveau racine apparaissent. La console montre les préfixes de niveau supérieur (`Private/`, `Development/` et `Finance/`) sous la forme de dossiers de niveau racine. La clé d'objet `s3-dg.pdf` n'a pas de préfixe et apparaît donc en tant qu'élément de niveau racine.

Résumé de la procédure détaillée

Dans cette procédure détaillée, vous créez un compartiment avec trois dossiers (`Private`, `Development` et `Finance`) dans celui-ci.

Vous avez deux utilisateurs, Alice et Bob. Vous souhaitez qu'Alice accède uniquement au dossier `Development` et que Bob accède uniquement au dossier `Finance`. Vous souhaitez garder le contenu du dossier `Private` privé. Dans la procédure pas à pas, vous gérez l'accès en créant des utilisateurs IAM (l'exemple utilise les noms d'utilisateur Alice et Bob) et en leur accordant les autorisations nécessaires.

IAM prend également en charge la création de groupes d'utilisateurs et l'octroi d'autorisations au niveau du groupe, qui s'appliquent à tous les utilisateurs du groupe. Cela vous aide à mieux gérer les autorisations. Pour cet exercice, Alice et Bob ont besoin de certaines autorisations communes. Vous allez donc créer un groupe nommé `Consultants` et ajouter Alice et Bob à ce groupe. Vous commencerez par octroyer des autorisations en attachant une stratégie de groupe à ce groupe. Ensuite, vous ajouterez des autorisations spécifiques aux utilisateurs en attachant des stratégies aux utilisateurs spécifiques.

> **Note**
>
> Cette démonstration utilise `companybucket` comme nom de compartiment, Alice et Bob comme utilisateurs IAM, et `Consultants` comme nom de groupe. Comme Amazon S3 exige que les noms de compartiment soient uniques à l'échelle mondiale, vous devez remplacer le nom du compartiment par un nom de votre création.

Préparation de la procédure détaillée

Dans cet exemple, vous utilisez vos Compte AWS informations d'identification pour créer des utilisateurs IAM. Initialement, ces utilisateurs n'ont aucune autorisation. Vous octroyez de façon incrémentielle des autorisations à ces utilisateurs pour leur permettre d'effectuer des actions Amazon S3 spécifiques. Pour tester ces autorisations, vous vous connectez à la console avec les autorisations de chaque utilisateur. Au fur et à mesure que vous accordez des autorisations en tant que Compte AWS propriétaire et que vous testez des autorisations en tant qu'utilisateur IAM, vous devez vous connecter et vous déconnecter en utilisant à chaque fois des informations d'identification différentes. Vous pouvez effectuer ce test dans un seul navigateur, mais le processus sera accéléré si vous pouvez utiliser deux navigateurs différents. Utilisez un navigateur pour vous connecter à l' AWS Management Console aide de vos Compte AWS informations d'identification et un autre navigateur pour vous connecter aux informations d'identification utilisateur IAM.

Pour vous connecter à l' AWS Management Console aide de vos Compte AWS informations d'identification, rendez-vous sur https://console.aws.amazon.com/. Un utilisateur IAM ne peut pas se

connecter en utilisant le même lien. Un utilisateur IAM doit utiliser une page de connexion prenant en charge IAM. En tant que propriétaire du compte, vous pouvez fournir ce lien à vos utilisateurs.

Pour plus d'informations sur IAM, consultez la page connexion à la AWS Management Console dans le Guide de l'utilisateur IAM.

Pour fournir un lien de connexion pour les utilisateurs IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau Navigation (Navigation), choisissez IAM Dashboard (Tableau de bord IAM).
3. Notez l'URL sous IAM users sign in link: (Lien de connexion des utilisateurs IAM). Vous donnerez ce lien aux utilisateurs IAM pour qu'ils se connectent à la console avec leur nom d'utilisateur et leur mot de passe IAM.

Étape 1 : Créer un compartiment

Au cours de cette étape, vous vous connectez à la console Amazon S3 avec vos Compte AWS informations d'identification, vous créez un compartiment, vous y ajoutez des dossiers et vous chargez un ou deux exemples de documents dans chaque dossier.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Créez un compartiment.

   Pour step-by-step obtenir des instructions, voirCréer un compartiment.
3. Chargez un document dans le compartiment.

   Cet exercice suppose que vous disposez du document `s3-dg.pdf` au niveau racine de ce compartiment. Si vous chargez un autre document, inscrivez son nom de fichier à la place de `s3-dg.pdf`.
4. Ajoutez trois dossiers nommés `Private`, `Finance` et `Development` au compartiment.

   Pour step-by-step obtenir des instructions sur la création d'un dossier, consultez Organisation des objets dans la console Amazon S3 à l'aide de dossiers > dans le guide de l'utilisateur d'Amazon Simple Storage Service.
5. Chargez un ou deux documents dans chaque dossier.

Pour cet exercice, supposons que vous avez chargé deux documents dans chaque dossier, de sorte que le compartiment possède des objets avec les clés suivantes :

- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

Pour step-by-step obtenir des instructions, voirChargement d'objets.

Étape 2 : Créer des utilisateurs IAM et un groupe

Utilisez maintenant la console IAM pour ajouter deux utilisateurs IAM, Alice et Bob, à votre. Compte AWS Pour step-by-step obtenir des instructions, reportez-vous à la section Création d'un utilisateur IAM Compte AWS dans votre guide de l'utilisateur IAM.

Créez également un groupe administratif nomméConsultants. Ajoutez ensuite les deux utilisateurs au groupe. Pour step-by-step obtenir des instructions, consultez la section Création de groupes d'utilisateurs IAM.

> **Warning**
>
> Lorsque vous ajoutez des utilisateurs et un groupe, n'attachez aucune stratégie octroyant des autorisations à ces utilisateurs. Au début, les utilisateurs n'ont aucune autorisation. Dans les sections suivantes, vous allez octroyer des autorisations de façon incrémentielle. Vous devez tout d'abord vous assurer d'avoir attribué des mots de passe à ces utilisateurs IAM. Vous utiliserez les autorisations de ces utilisateurs pour tester les actions Amazon S3 et vérifier que ces autorisations fonctionnent comme prévu.

Pour step-by-step obtenir des instructions relatives à la création d'un nouvel utilisateur IAM, reportez-vous à la section Création d'un utilisateur IAM Compte AWS dans votre guide de l'utilisateur IAM.

Lorsque vous créez les utilisateurs pour cette procédure, sélectionnez Accès àAWS Management Console et désactivez l'option Accès par programmation.

Pour step-by-step obtenir des instructions sur la création d'un groupe administratif, consultez la section Création de votre premier utilisateur et de votre premier groupe administrateur IAM dans le guide de l'utilisateur IAM.

Étape 3 : Vérifier que les utilisateurs IAM ne disposent d'aucune autorisation

Si vous utilisez deux navigateurs, vous pouvez utiliser à présent le second navigateur pour vous connecter à la console en utilisant les autorisations d'un des utilisateurs IAM.

1. Utilisez le lien de connexion d'utilisateur IAM (consultez Pour fournir un lien de connexion pour les utilisateurs IAM) pour vous connecter à la AWS Management Console en utilisant les autorisations d'un des deux utilisateurs IAM.
2. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

   Vérifiez le message de console indiquant que l'accès est refusé.

À présent, vous pouvez commencer octroyer des autorisations incrémentielles aux utilisateurs. En premier lieu, vous attacherez une stratégie de groupe octroyant les autorisations que les deux utilisateurs doivent posséder.

Étape 4 : Octroyer des autorisations au niveau du groupe

Vous voulez que les utilisateurs soient en mesure d'effectuer les tâches suivantes :

- Répertorier tous les compartiments détenus par le compte parent. Pour cela, Bob et Alice doivent disposer de l'autorisation pour l'action `s3:ListAllMyBuckets`.
- Répertorier les éléments de niveau racine, les dossiers et les objets dans le compartiment `companybucket`. Pour cela, Bob et Alice doivent disposer de l'autorisation pour l'action `s3:ListBucket` sur le compartiment `companybucket`.

À présent, vous allez créer une stratégie qui octroie ces autorisations, puis vous l'attacherez au groupe `Consultants`.

Étape 4.1 : Octroyer l'autorisation de répertorier tous les compartiments

Dans cette étape, vous allez créer une stratégie gérée qui octroie aux utilisateurs les autorisations minimales leur permettant de répertorier tous les compartiments détenus par le compte parent. Vous attacherez ensuite cette stratégie au groupe `Consultants`. Lorsque vous attachez la stratégie gérée à un utilisateur ou à un groupe, vous accordez à cet utilisateur ou à ce groupe l'autorisation d'obtenir la liste des compartiments détenus par le parent Compte AWS.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

   > **Note**
   >
   > Comme vous octroyez des autorisations utilisateur, connectez-vous à l'aide des informations d'identification de votre Compte AWS , et non pas en tant qu'utilisateur IAM.

2. Créez la stratégie gérée.

   a. Dans le volet de navigation à gauche, choisissez Stratégies, puis Créer une stratégie.

   b. Choisissez l'onglet JSON.

   c. Copiez la stratégie d'accès suivante et collez-la dans le champ de texte de stratégie.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "AllowGroupToSeeBucketListInTheConsole",
         "Action": ["s3:ListAllMyBuckets"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::*"]
       }
     ]
   }
   ```

   Une stratégie est un document JSON. Dans ce document, un élément `Statement` est un tableau d'objets, chacun décrivant une autorisation à l'aide d'une collection de paires de noms-valeurs. La stratégie précédente décrit une autorisation spécifique. L'élément `Action` spécifie le type d'accès. Dans la stratégie, l'élément `s3:ListAllMyBuckets` est une action Amazon S3 prédéfinie. Cette action couvre l'opération Amazon S3 GET Service, qui renvoie

une liste de tous les buckets détenus par l'expéditeur authentifié. La valeur de l'élément `Effect` détermine si une autorisation spécifique est accordée ou refusée.

d. Choisissez Review Policy (Examiner une stratégie). Sur la page suivante, saisissez `AllowGroupToSeeBucketListInTheConsole`dans le champ Nom, puis choisissez Créer une stratégie.

> **Note**
>
> L'entrée Résumé affiche un message indiquant que la stratégie n'accorde pas d'autorisations. Vous pouvez ignorer ce message dans le cadre de cette procédure.

3. Attachez la stratégie gérée `AllowGroupToSeeBucketListInTheConsole` que vous avez créée au groupe `Consultants`.

   Pour step-by-step obtenir des instructions relatives à l'attachement d'une politique gérée, consultez la section Ajout et suppression d'autorisations d'identité IAM dans le guide de l'utilisateur IAM.

   Vous attachez des documents de stratégie aux utilisateurs IAM et aux groupes dans la console IAM. Comme vous voulez que les deux utilisateurs soient en mesure de répertorier les compartiments, vous attachez cette stratégie au groupe.

4. Testez l'autorisation.

   a. Utilisez le lien de connexion d'utilisateur IAM (consultez Pour fournir un lien de connexion pour les utilisateurs IAM) pour vous connecter à la console en utilisant les autorisations d'un des deux utilisateurs IAM.

   b. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

      La console doit à présent répertorier tous les compartiments mais pas les objets figurant dans ces compartiments.

Étape 4.2 : Permettre aux utilisateurs de répertorier le contenu de niveau racine d'un compartiment

Ensuite, autorisez tous les utilisateurs du groupe `Consultants` à répertorier les éléments de niveau racine du compartiment `companybucket`. Lorsqu'un utilisateur choisit le compartiment de l'entreprise dans la console Amazon S3, il peut voir les éléments de niveau racine figurant dans le compartiment.

> **Note**
>
> Cet exemple utilise `companybucket` à titre d'illustration. Vous devez utiliser le nom du compartiment que vous avez créé.

Pour comprendre la demande que la console envoie à Amazon S3 lorsque vous choisissez un nom de compartiment, la réponse renvoyée par Amazon S3 et la façon dont la console interprète la réponse, examinez le flux d'un peu plus près.

Lorsque vous choisissez un nom de compartiment, la console envoie la demande GET Bucket (List Objects) à Amazon S3. Cette demande inclut les paramètres suivants :

- le paramètre `prefix` avec une chaîne vide comme valeur ;
- le paramètre `delimiter` avec `/` comme valeur.

Voici un exemple de demande.

```
GET ?prefix=&delimiter=/ HTTP/1.1
Host: companybucket.s3.amazonaws.com
Date: Wed, 01 Aug  2012 12:00:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

Amazon S3 retourne une réponse qui inclut l'élément `<ListBucketResult/>`.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix></Prefix>
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>s3-dg.pdf</Key>
    ...
  </Contents>
  <CommonPrefixes>
    <Prefix>Development/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Finance/</Prefix>
  </CommonPrefixes>
```

```
  <CommonPrefixes>
    <Prefix>Private/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

L'objet de clé `s3-dg.pdf` ne contient pas la barre oblique (`/`) comme délimiteur et Amazon S3 retourne la clé dans l'élément `<Contents>`. Toutefois, toutes les autres clés dans notre exemple de compartiment contiennent le délimiteur `/`. Amazon S3 regroupe ces clés et renvoie un élément `<CommonPrefixes>` pour chacune des valeurs de préfixe distinctes `Development/`, `Finance/` et `Private/` qui est une sous-chaîne du début de ces clés jusqu'à la première occurrence du délimiteur `/` spécifié.

La console interprète ce résultat et affiche les éléments de niveau racine sous la forme de trois dossiers et d'une clé d'objet.

Si Bob ou Alice ouvre le dossier Development (Développement), la console envoie la demande GET Bucket (List Objects) à Amazon S3 avec le préfixe `prefix` et les valeurs suivantes pour les paramètres `delimiter` :

- Le paramètre `prefix` avec la valeur `Development/`.
- Le paramètre `delimiter` avec la valeur « `/` ».

En réponse à cela, Amazon S3 renvoie les clés d'objet qui commencent par le préfixe spécifié.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix>Development</Prefix>
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>Project1.xls</Key>
    ...
  </Contents>
  <Contents>
    <Key>Project2.xls</Key>
    ...
  </Contents>
</ListBucketResult>
```

La console affiche les clés d'objet.

A présent, revenez à l'octroi aux utilisateurs d'une autorisation pour répertorier les éléments de niveau racine du compartiment. Pour répertorier le contenu du compartiment, les utilisateurs ont besoin d'une autorisation pour appeler l'action `s3:ListBucket`, telle qu'illustrée dans la déclaration de stratégie ci-dessous. Pour s'assurer qu'ils ne voient que le contenu de niveau racine, vous ajoutez une condition stipulant que les utilisateurs doivent spécifier un paramètre `prefix` vide dans la demande (c'est-à-dire qu'ils ne sont pas autorisés à double-cliquer sur les dossiers de niveau racine). Enfin, vous ajoutez une condition pour exiger un accès de type dossier en exigeant que les demandes des utilisateurs incluent le paramètre `delimiter` avec la valeur « `/` ».

```
{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
         "StringEquals":{
             "s3:prefix":[""], "s3:delimiter":["/"]
                       }
              }
}
```

Lorsque vous choisissez un compartiment sur la console Amazon S3, la console envoie d'abord la demande de localisation du compartiment GET pour savoir Région AWS où le compartiment est déployé. Ensuite, la console utilise le point de terminaison spécifique à la Région du compartiment pour envoyer la demande GET Bucket (List Objects). Par conséquent, si les utilisateurs doivent utiliser la console, vous devez octroyer l'autorisation pour l'action `s3:GetBucketLocation` telle qu'illustrée dans la déclaration de stratégie suivante.

```
{
   "Sid": "RequiredByS3Console",
   "Action": ["s3:GetBucketLocation"],
   "Effect": "Allow",
   "Resource": ["arn:aws:s3:::*"]
}
```

Pour permettre aux utilisateurs de répertorier le contenu de niveau racine d'un compartiment

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

Utilisez vos Compte AWS informations d'identification, et non celles d'un utilisateur IAM, pour vous connecter à la console.

2. Remplacez la stratégie gérée `AllowGroupToSeeBucketListInTheConsole` existante qui est attachée au groupe `Consultants` par la stratégie suivante, qui autorise également l'action `s3:ListBucket`. N'oubliez pas de remplacer *`companybucket`*dans la politique `Resource` par le nom de votre compartiment.

   Pour step-by-step obtenir des instructions, consultez la section Modification des politiques IAM dans le Guide de l'utilisateur IAM. Lorsque vous suivez les step-by-step instructions, veillez à suivre les étapes permettant d'appliquer vos modifications à toutes les entités principales auxquelles la politique est attachée.

```
{
  "Version": "2012-10-17",
  "Statement": [
     {
       "Sid":
 "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
       "Action": [ "s3:ListAllMyBuckets", "s3:GetBucketLocation" ],
       "Effect": "Allow",
       "Resource": [ "arn:aws:s3:::*"  ]
     },
     {
       "Sid": "AllowRootLevelListingOfCompanyBucket",
       "Action": ["s3:ListBucket"],
       "Effect": "Allow",
       "Resource": ["arn:aws:s3:::companybucket"],
       "Condition":{
             "StringEquals":{
                    "s3:prefix":[""], "s3:delimiter":["/"]
                           }
                  }
     }
  ]
}
```

3. Testez les autorisations mises à jour.

   a. Utilisez le lien de connexion d'utilisateur IAM (voir Pour fournir un lien de connexion pour les utilisateurs IAM) pour vous connecter à la AWS Management Console.

Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

b. Choisissez le compartiment que vous avez créé ; la console affiche les éléments de niveau racine du compartiment. Si vous choisissez un dossier dans le compartiment, vous ne pourrez pas voir le contenu du dossier, car vous n'avez pas encore été octroyé ces autorisations.

Ce test réussit lorsque les utilisateurs utilisent la console Amazon S3. Lorsque vous choisissez sur un compartiment dans la console, l'implémentation de la console envoie une demande qui inclut le paramètre `prefix` avec une chaîne vide comme valeur et le paramètre `delimiter` avec la valeur « `/` ».

Étape 4.3 : Récapitulatif de la stratégie de groupe

L'effet net de la stratégie de groupe que vous avez ajoutée est d'octroyer aux utilisateurs IAM Alice et Bob les autorisations minimales suivantes :

- Répertorier tous les compartiments détenus par le compte parent.
- Voir les éléments de niveau racine dans le compartiment `companybucket`

Toutefois, les utilisateurs ont une marge de manœuvre encore limitée. Vous octroyez ensuite des autorisations spécifiques à l'utilisateur, comme suit :

- Autorisez Alice à obtenir et placer des objets dans le dossier `Development`.
- Autorisez Bob à obtenir et placer des objets dans le dossier `Finance`.

Dans le cadre des autorisations spécifiques à l'utilisateur, vous attachez une stratégie à un utilisateur spécifique et non pas au groupe. Dans la section suivante, vous octroyez à Alice l'autorisation de travailler dans le dossier `Development`. Vous pouvez répéter cette procédure pour octroyer une autorisation similaire à Bob pour lui permettre de travailler dans le dossier `Finance`.

Étape 5 : Octroyer des autorisations spécifiques à l'utilisateur IAM Alice

À présent, vous octroyez des autorisations supplémentaires à Alice pour qu'elle puisse voir le contenu du dossier `Development`, ainsi qu'obtenir et placer des objets dans ce dossier.

Étape 5.1 : Octroyer une autorisation à l'utilisateur IAM Alice pour répertorier le contenu du dossier Development

Pour qu'Alice répertorie le contenu du `Development` dossier, vous devez appliquer à l'utilisateur Alice une politique qui autorise l'`s3:ListBucket`action sur le `companybucket` compartiment, à condition que la demande inclut le préfixe`Development/`. Vous voulez que cette stratégie soit appliquée uniquement à l'utilisateur Alice. Vous devez donc une stratégie en ligne. Pour de plus amples informations sur les politiques en ligne, consultez Politiques gérées et politiques en ligne dans le Guide de l'utilisateur IAM.

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

   Utilisez vos Compte AWS informations d'identification, et non celles d'un utilisateur IAM, pour vous connecter à la console.

2. Créez une stratégie en ligne pour octroyer à l'utilisateur Alice l'autorisation de répertorier le contenu du dossier `Development`.

   a. Dans le panneau de navigation de gauche, choisissez Utilisateurs.

   b. Choisissez le nom d'utilisateur Alice.

   c. Sur la page des détails de l'utilisateur, choisissez l'onglet Autorisations, puis Ajouter une stratégie en ligne.

   d. Choisissez l'onglet JSON.

   e. Copiez la politique suivante et collez-la dans le champ de texte de la politique.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{  "StringLike":{"s3:prefix":["Development/*"] }
       }
    }
  ]
}
```

f. Choisissez Review Policy (Examiner une stratégie). Sur la page suivante, saisissez un nom dans le champ Nom, puis choisissez Créer une stratégie.

3. Testez la modification des autorisations d'Alice :

a. Utilisez le lien de connexion d'utilisateur IAM (voir Pour fournir un lien de connexion pour les utilisateurs IAM) pour vous connecter à la AWS Management Console.

b. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

c. Dans la console Amazon S3, vérifiez qu'Alice peut voir la liste des objets dans le dossier `Development/` du compartiment.

Lorsque l'utilisateur choisit le dossier `/Development` pour afficher la liste des objets qu'il contient, la console Amazon S3 envoie la demande `ListObjects` à Amazon S3 avec le préfixe `/Development`. Comme l'utilisateur obtient l'autorisation de voir la liste des objets avec le préfixe `Development` et le délimiteur `/`, Amazon S3 renvoie la liste des objets avec le préfixe de clé `Development/`, et la console affiche cette liste.

Étape 5.2 : Octroyer des autorisations à l'utilisateur IAM Alice pour obtenir et placer des objets dans le dossier Development

Pour qu'Alice puisse obtenir et placer des objets dans le dossier `Development`, elle a besoin d'une autorisation pour appeler les actions `s3:GetObject` et `s3:PutObject`. Les déclarations de stratégie suivantes octroient ces autorisations, à condition que la demande inclue le paramètre `prefix` avec la valeur `Development/`.

```
{
    "Sid":"AllowUserToReadWriteObjectData",
    "Action":["s3:GetObject", "s3:PutObject"],
    "Effect":"Allow",
    "Resource":["arn:aws:s3:::companybucket/Development/*"]
 }
```

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

Utilisez vos Compte AWS informations d'identification, et non celles d'un utilisateur IAM, pour vous connecter à la console.

2. Modifiez la stratégie en ligne que vous avez créée à l'étape précédente.

a. Dans le panneau de navigation de gauche, choisissez Utilisateurs.

b. Choisissez sur le nom d'utilisateur Alice.

c. Dans la page des détails, sélectionnez l'onglet Autorisations et développez la section Stratégies en ligne.

d. En regard du nom de la stratégie que vous avez créée à l'étape précédente, choisissez Modifier la stratégie.

e. Copiez la stratégie suivante et collez-la dans le champ de texte de la stratégie pour remplacer la stratégie existante.

```
{
     "Version": "2012-10-17",
     "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      }
   ]
}
```

3. Testez la stratégie mise à jour :

a. Utilisez le lien de connexion d'utilisateur IAM (voir Pour fournir un lien de connexion pour les utilisateurs IAM) pour vous connecter à la AWS Management Console.

b. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

c. Dans la console Amazon S3, vérifiez qu'Alice peut à présent ajouter un objet et télécharger un objet dans le dossier `Development`.

Étape 5.3 : Refuser explicitement à l'utilisateur IAM Alice les autorisations pour tout autre dossier du compartiment

L'utilisateur Alice peut à présent répertorier le contenu de niveau racine dans le compartiment `companybucket`. Elle peut également obtenir et placer des objets dans le dossier `Development`. Si vous souhaitez réellement renforcer les autorisations d'accès, vous pouvez refuser explicitement à Alice l'accès aux autres dossiers du compartiment. S'il existe une autre stratégie (stratégie de compartiment ou liste ACL) qui accorde à Alice l'accès à d'autres dossiers du compartiment, ce refus explicite a priorité sur ces autorisations.

Vous pouvez ajouter la déclaration suivante à la stratégie de l'utilisateur Alice. Elle exige que toutes les demandes envoyées par Alice à Amazon S3 incluent le paramètre `prefix`, dont la valeur peut être `Development/*` ou une chaîne vide.

```
{
   "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
   "Action": ["s3:ListBucket"],
   "Effect": "Deny",
   "Resource": ["arn:aws:s3:::companybucket"],
   "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                  "Null"         : {"s3:prefix":false }
    }
}
```

Deux expressions conditionnelles figurent dans le bloc `Condition`. Les résultats de ces expressions conditionnelles sont associés à l'aide du `AND` logique. Si les deux conditions son vraies, le résultat de la combinaison combinée est vrai (true). Comme `Effect` dans cette stratégie est défini sur `Deny`, lorsque la `Condition` a la valeur true, les utilisateurs ne peuvent pas effectuer l'`Action` spécifiée.

- L'expression conditionnelle `Null` garantit que les demandes d'Alice incluent le paramètre `prefix`.

  Le paramètre `prefix` requiert un accès de type dossier. Si vous envoyez une demande sans le paramètre `prefix`, Amazon S3 renvoie toutes les clés d'objet.

  Si la demande inclut le paramètre `prefix` avec une valeur null, l'expression prend la valeur true, et le bloc entier `Condition` est défini sur true. Vous devez autoriser une chaîne vide comme valeur du paramètre `prefix`. Nous avons vu dans la discussion précédente qu'autoriser la chaîne null permet à Alice de récupérer les éléments de niveau racine du compartiment, comme le fait la

console dans la discussion précédente. Pour plus d'informations, consultez Étape 4.2 : Permettre aux utilisateurs de répertorier le contenu de niveau racine d'un compartiment.

- L'expression conditionnelle `StringNotLike` garantit que si la valeur du paramètre `prefix` est spécifiée et n'est pas `Development/*`, la demande échoue.

Suivez les étapes de la section précédente et mettez à jour à nouveau la stratégie en ligne que vous avez créée pour l'utilisateur Alice.

Copiez la stratégie suivante et collez-la dans le champ de texte de la stratégie pour remplacer la stratégie existante.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      },
      {
         "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
         "Action": ["s3:ListBucket"],
         "Effect": "Deny",
         "Resource": ["arn:aws:s3:::companybucket"],
         "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                        "Null"         : {"s3:prefix":false }
          }
      }
   ]
}
```

Étape 6 : Octroyer des autorisations spécifiques à l'utilisateur IAM Bob

À présent, vous voulez accorder à Bob une autorisation sur le dossier `Finance`. Suivez la procédure que vous avez utilisée précédemment pour accorder des autorisations à Alice, mais remplacez le dossier `Development` par le dossier `Finance`. Pour step-by-step obtenir des instructions, voirÉtape 5 : Octroyer des autorisations spécifiques à l'utilisateur IAM Alice.

Étape 7 : Sécuriser le dossier Private

Dans cet exemple, vous n'avez que deux utilisateurs. Vous avez accordé toutes les autorisations minimales requises au niveau du groupe et accordé des autorisations au niveau de l'utilisateur seulement lorsque ces autorisations étaient réellement requises au niveau d'un utilisateur individuel. Cette approche aide à réduire au maximum l'effort de gestion des autorisations. Lorsque le nombre d'utilisateurs augmente, la gestion des autorisations peut devenir fastidieuse. Par exemple, vous ne voulez pas que les utilisateurs de cet exemple accèdent au contenu du dossier `Private`. Comment vous assurer que vous n'accordez pas accidentellement à un utilisateur l'autorisation d'accéder au `Private` dossier ? Vous devez ajouter une stratégie qui refuse explicitement l'accès à ce dossier. Un refus explicite a priorité sur toutes les autres autorisations.

Pour vous assurer que le dossier `Private` reste privé, vous pouvez ajouter les deux déclarations de refus suivantes à la stratégie de groupe :

- Ajoutez la déclaration suivante pour refuser explicitement toute action sur les ressources dans le dossier `Private` (`companybucket/Private/*`).

```
{
  "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
  "Action": ["s3:*"],
  "Effect": "Deny",
  "Resource":["arn:aws:s3:::companybucket/Private/*"]
}
```

- Vous refusez également l'autorisation pour l'action visant à répertorier les objets lorsque la demande spécifie le préfixe `Private/`. Dans la console, si Bob ou Alice ouvre le dossier `Private`, avec cette stratégie, Amazon S3 renvoie une réponse d'erreur.

```
{
  "Sid": "DenyListBucketOnPrivateFolder",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
```

```
  "Condition":{
      "StringLike":{"s3:prefix":["Private/"]}
   }
}
```

Remplacez la stratégie de groupe `Consultants` par une stratégie mise à jour qui inclut les déclarations de refus précédentes. Après l'application de la stratégie mise à jour, aucun des utilisateurs du groupe ne peut accéder au dossier `Private` de votre compartiment.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   Utilisez vos Compte AWS informations d'identification, et non celles d'un utilisateur IAM, pour vous connecter à la console.

2. Remplacez la stratégie gérée `AllowGroupToSeeBucketListInTheConsole` existante qui est attachée au groupe `Consultants` par la stratégie suivante. N'oubliez pas de remplacer *`companybucket`* dans la stratégie par le nom de votre compartiment.

   Pour obtenir des instructions, consultez la section Modification des politiques gérées par le client dans le guide de l'utilisateur IAM. Lorsque vous suivez ces instructions, veillez à suivre les consignes pour appliquer vos modifications à toutes les entités principales auxquelles la stratégie est attachée.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid":
    "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
         "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::*"]
       },
       {
         "Sid": "AllowRootLevelListingOfCompanyBucket",
         "Action": ["s3:ListBucket"],
         "Effect": "Allow",
         "Resource": ["arn:aws:s3:::companybucket"],
         "Condition":{
             "StringEquals":{"s3:prefix":[""]}
   ```

```
        }
      },
      {
        "Sid": "RequireFolderStyleList",
        "Action": ["s3:ListBucket"],
        "Effect": "Deny",
        "Resource": ["arn:aws:s3:::*"],
        "Condition":{
            "StringNotEquals":{"s3:delimiter":"/"}
        }
       },
      {
        "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
        "Action": ["s3:*"],
        "Effect": "Deny",
        "Resource":["arn:aws:s3:::companybucket/Private/*"]
      },
      {
        "Sid": "DenyListBucketOnPrivateFolder",
        "Action": ["s3:ListBucket"],
        "Effect": "Deny",
        "Resource": ["arn:aws:s3:::*"],
        "Condition":{
            "StringLike":{"s3:prefix":["Private/"]}
        }
      }
    ]
}
```

Étape 8 : Nettoyage

Pour faire le ménage, ouvrez la console IAM et supprimez les utilisateurs Alice et Bob. Pour step-by-step obtenir des instructions, reportez-vous à la section Suppression d'un utilisateur IAM dans le guide de l'utilisateur IAM.

Pour vous assurer que le stockage ne vous est plus facturé, vous devez également supprimer les objets et le compartiment que vous avez créés pour cet exercice.

Ressources connexes

- Gestion des politiques IAM dans le Guide de l'utilisateur IAM

Procédures pas à pas utilisant des politiques pour gérer l'accès à vos ressources Amazon S3

Cette rubrique fournit les exemples de procédures suivants permettant l'accès aux ressources d'Amazon S3. Ces exemples utilisent le AWS Management Console pour créer des ressources (compartiments, objets, utilisateurs) et leur accorder des autorisations. Ces exemples montrent ensuite comment vérifier les autorisations à l'aide des outils de ligne de commande afin de vous éviter d'avoir à écrire un code. Nous fournissons des commandes en utilisant à la fois le AWS Command Line Interface (AWS CLI) et le AWS Tools for Windows PowerShell.

- Exemple 1 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations sur un compartiment

  Par défaut, les utilisateurs IAM que vous créez dans votre compte n'ont pas d'autorisation. Dans cet exercice, vous accordez une autorisation à un utilisateur pour exécuter des opérations de compartiments et d'objets.

- Exemple 2 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations entre comptes sur un compartiment

  Dans cet exercice, un propriétaire de compartiment, le compte A, accorde des autorisations d'accès inter-comptes à un autre Compte AWS, le compte B. Le compte B délègue ensuite ces autorisations aux utilisateurs dans son compte.

- Gestion des autorisations lorsque les propriétaires d'objets et de compartiments ne sont pas les mêmes

  Ces exemples de scénario décrivent un propriétaire de compartiment concédant des autorisations d'objets à d'autres, toutefois tous les objets du compartiment n'appartiennent pas à son propriétaire. De quelles autorisations le propriétaire du compartiment a besoin et comment peut-il déléguer ces autorisations ?

  Celui Compte AWS qui crée un bucket s'appelle le propriétaire du bucket. Le propriétaire peut accorder d'autres Comptes AWS autorisations pour télécharger des objets, et ceux Comptes AWS qui créent des objets en sont propriétaires. Le propriétaire du compartiment n'a aucune autorisation sur ces objets créés par d'autres Comptes AWS. Si le propriétaire du compartiment rédige une politique de compartiment accordant l'accès aux objets, cette politique ne s'applique pas aux objets appartenant à d'autres comptes.

Dans ce cas, le propriétaire de l'objet doit d'abord accorder des autorisations au propriétaire du compartiment utilisant une liste ACL d'objet. Le propriétaire du bucket peut ensuite déléguer ces autorisations d'objets à d'autres personnes, à des utilisateurs de son propre compte ou à un autre Compte AWS, comme l'illustrent les exemples suivants.

- Exemple 3 : propriétaire d'un compartiment accordant des autorisations sur des objets qu'il ne possède pas

  Dans cet exercice, le propriétaire du compartiment obtient d'abord les autorisations du propriétaire de l'objet. Le propriétaire du compartiment délègue ensuite ces autorisations aux utilisateurs dans son propre compte.

- Exemple 4 - Le propriétaire du bucket accorde une autorisation multicompte à des objets qu'il ne possède pas

  Après avoir reçu les autorisations du propriétaire de l'objet, celui-ci ne peut pas déléguer d'autorisation à d'autres personnes Comptes AWS car la délégation entre comptes n'est pas prise en charge (voirDélégation d'autorisations). Au lieu de cela, le propriétaire du compartiment peut créer un rôle IAM autorisé à effectuer des opérations spécifiques (telles que obtenir un objet) et autoriser un autre Compte AWS à assumer ce rôle. Toute personne occupant ce rôle peut ensuite accéder aux objets. Cet exemple montre comment un propriétaire du compartiment peut utiliser un rôle IAM pour permettre la délégation entre comptes.

Avant d'essayer les procédures d'exemples

Ces exemples utilisent le AWS Management Console pour créer des ressources et accorder des autorisations. Pour tester les autorisations, les exemples utilisent les outils de ligne de commande AWS CLI AWS Tools for Windows PowerShell, et vous n'avez donc pas besoin d'écrire de code. Afin de tester des autorisations, vous devez installer un de ces outils. Pour plus d'informations, consultez Configuration des outils pour les procédures pas à pas.

En outre, lors de la création de ressources, ces exemples n'utilisent pas les informations d'identification de l'utilisateur root d'un Compte AWS. A la place, vous créez un administrateur pour ces comptes afin qu'il exécute ces tâches.

Utilisation d'un utilisateur administrateur pour créer des ressources et accorder des autorisations

AWS Identity and Access Management (IAM) déconseille l'utilisation des informations d'identification d'utilisateur root de votre Compte AWS pour effectuer des demandes. Il est préférable de créer un rôle ou un utilisateur IAM, d'accorder un accès total à cet utilisateur et d'utiliser ses informations

d'identification pour effectuer des demandes. Nous l'appelons rôle ou utilisateur administrateur. Pour plus d'informations, consultez Informations d'identification Utilisateur racine d'un compte AWS et identités IAM dans Références générales AWS et dans Bonnes pratiques IAM du Guide de l'utilisateur IAM.

Toutes les exemples de démonstration dans cette section utilisent les autorisations de l'utilisateur administrateur. Si vous n'avez pas créé d'utilisateur administrateur pour votre compte Compte AWS, les rubriques vous indiquent comment procéder.

Pour vous connecter à l' AWS Management Console aide des informations d'identification de l'utilisateur, vous devez utiliser l'URL de connexion de l'utilisateur IAM. La console IAM fournit cette URL pour votre Compte AWS. Les rubriques vous indiquent comment obtenir cette URL.

Configuration des outils pour les procédures pas à pas

Les exemples d'introduction (voirProcédures pas à pas utilisant des politiques pour gérer l'accès à vos ressources Amazon S3) utilisent le AWS Management Console pour créer des ressources et accorder des autorisations. Pour tester les autorisations, les exemples utilisent les outils de ligne de commande AWS Command Line Interface (AWS CLI) et AWS Tools for Windows PowerShell vous n'avez donc pas besoin d'écrire de code. Afin de tester des autorisations, vous devez installer un de ces outils.

Pour configurer le AWS CLI

1. Téléchargez et configurez l'interface AWS CLI. Pour obtenir des instructions, consultez les rubriques suivantes dans le Guide de l'utilisateur de l'interface AWS Command Line Interface :

   Installez ou mettez à jour vers la dernière version du AWS Command Line Interface

   Commencez avec le AWS Command Line Interface
2. Définir le profil par défaut.

   Vous stockez les informations d'identification de l'utilisateur dans le fichier de AWS CLI configuration. Créez un profil par défaut dans le fichier de configuration à l'aide de vos Compte AWS informations d'identification. Pour obtenir des instructions sur la recherche et la modification de votre fichier de AWS CLI configuration, consultez la section Paramètres du fichier de configuration et d'identification.

   ```
   [default]
   aws_access_key_id = access key ID
   ```

```
aws_secret_access_key = secret access key
region = us-west-2
```

3. Vérifiez la configuration en saisissant la commande suivante à l'invite de commande. Ces deux commandes ne fournissent pas directement d'autorisations, par conséquent ce sont les informations du profil par défaut qui sont utilisées.

   - Essayez la `help` commande.

     ```
     aws help
     ```

   - Pour obtenir la liste des buckets du compte configuré, utilisez la `aws s3 ls` commande.

     ```
     aws s3 ls
     ```

Au fur et à mesure des procédures pas à pas, vous créerez des utilisateurs et vous enregistrerez les informations d'identification des utilisateurs dans les fichiers de configuration en créant des profils, comme le montre l'exemple suivant. Ces profils portent les noms de `AccountAadmin` et`AccountBadmin`.

```
[profile AccountAadmin]
aws_access_key_id = User AccountAadmin access key ID
aws_secret_access_key = User AccountAadmin secret access key
region = us-west-2

[profile AccountBadmin]
aws_access_key_id = Account B access key ID
aws_secret_access_key = Account B secret access key
region = us-east-1
```

Pour exécuter une commande avec ces informations d'identification utilisateur, ajoutez un paramètre `--profile` spécifiant le nom du profil. La AWS CLI commande suivante permet de récupérer une liste d'objets *`examplebucket`*et de spécifier le `AccountBadmin` profil.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

Vous pouvez aussi configurer une série d'informations d'identification utilisateur en tant que profil par défaut en modifiant la variable d'environnement `AWS_DEFAULT_PROFILE` à l'invite de commande.

Ensuite, chaque fois que vous exécutez des AWS CLI commandes sans le `--profile` paramètre, le AWS CLI profil que vous avez défini dans la variable d'environnement est le profil par défaut.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Pour configurer AWS Tools for Windows PowerShell

1. Téléchargez et configurez l'interface AWS Tools for Windows PowerShell. Pour obtenir des instructions, reportez-vous à la section Installation du AWS Tools for Windows PowerShell dans le guide de AWS Tools for Windows PowerShell l'utilisateur.

   > **Note**
   >
   > Pour charger le AWS Tools for Windows PowerShell module, vous devez activer l'exécution PowerShell du script. Pour plus d'informations, consultez la section Activer l'exécution de scripts dans le guide de AWS Tools for Windows PowerShell l'utilisateur.

2. Pour ces procédures pas à pas, vous spécifiez les AWS informations d'identification par session à l'aide de la `Set-AWSCredentials` commande. La commande enregistre les informations d'identification dans un stockage permanent (paramètre `-StoreAs`).

   ```
   Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -
   storeas string
   ```

3. Vérifier la configuration

   - Pour récupérer la liste des commandes disponibles que vous pouvez utiliser pour les opérations Amazon S3, exécutez la `Get-Command` commande.

     ```
     Get-Command -module awspowershell -noun s3* -StoredCredentials string
     ```

   - Pour récupérer la liste des objets d'un bucket, exécutez la `Get-S3Object` commande.

     ```
     Get-S3Object -BucketName bucketname -StoredCredentials string
     ```

Pour obtenir la liste des commandes, consultez la section AWS Outils de référence pour les PowerShell applets de commande.

Vous êtes maintenant prêt à essayer les procédures pas à pas. Suivez les liens fournis au début de chaque section.

Exemple 1 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations sur un compartiment

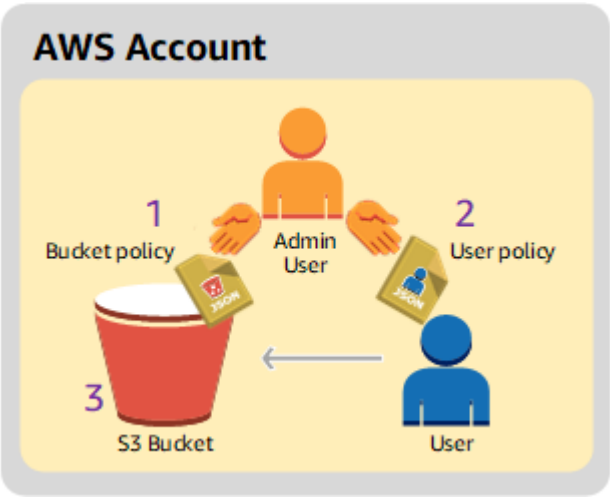
> **Important**
>
> Il est préférable d'accorder des autorisations aux rôles IAM plutôt que d'accorder des autorisations à des utilisateurs individuels. Pour plus d'informations sur la façon d'accorder des autorisations aux rôles IAM, consultez. Comprendre les autorisations entre comptes et utiliser les rôles IAM

Rubriques

- Préparation de la procédure détaillée
- Étape 1 : créer des ressources dans le compte A et accorder des autorisations
- Étape 2 : testez les autorisations

Dans cette procédure pas à pas, un utilisateur Compte AWS possède un bucket et le compte inclut un utilisateur IAM. Par défaut, l'utilisateur n'a aucune autorisation. Le compte parent doit accorder des autorisations à l'utilisateur pour que celui-ci puisse exécuter des tâches. Le propriétaire du compartiment et le compte parent sont identiques. Par conséquent, pour accorder à l'utilisateur des autorisations sur le compartiment, il Compte AWS peut utiliser une politique de compartiment, une politique utilisateur ou les deux. Le propriétaire du compte accorde certaines autorisations grâce à une stratégie de compartiment et d'autres autorisations grâce à une stratégie d'utilisateur.

Les étapes suivantes résument la procédure :



1. L'administrateur du compte crée une stratégie de compartiment qui accorde un ensemble d'autorisations à l'utilisateur.
2. L'administrateur du compte attache une stratégie d'utilisateur à l'utilisateur en accordant des autorisations supplémentaires.
3. L'utilisateur essaie ensuite les autorisations accordées via la stratégie de compartiment et utilisateur.

Pour cet exemple, vous aurez besoin d'un Compte AWS. Au lieu d'utiliser les informations d'identification d'utilisateur root du compte, vous créez un utilisateur administrateur (consultez Utilisation d'un utilisateur administrateur pour créer des ressources et accorder des autorisations). Nous faisons référence à l'utilisateur Compte AWS et à l'utilisateur administrateur comme indiqué dans le tableau suivant.

ID de compte	Compte désigné comme	Utilisateur administrateur du compte
1111-1111-1111	Compte A	AccountAadmin

> **Note**
>
> Dans cet exemple, l'utilisateur administrateur est AccountAadmin, ce qui fait référence au compte A, et non AccountAdmin.

Toutes les tâches de création d'utilisateurs et d'octroi d'autorisations sont effectuées dans la AWS Management Console. Pour vérifier les autorisations, la procédure pas à pas utilise les outils de ligne de commande AWS Command Line Interface (AWS CLI) et AWS Tools for Windows PowerShell vous n'avez donc pas besoin d'écrire de code.

Préparation de la procédure détaillée

1. Assurez-vous que vous disposez d'un Compte AWS et qu'il possède un utilisateur doté de privilèges d'administrateur.

   a. Inscrivez-vous pour un Compte AWS, si nécessaire. Nous appelons ce compte : Compte A.

      i. Accédez à https://aws.amazon.com/s3 et choisissez Créer un AWS compte.

ii. Suivez les instructions à l'écran.

AWS vous informera par e-mail lorsque votre compte sera actif et que vous pourrez l'utiliser.

b. Dans le compte A, créez un utilisateur administrateur**`AccountAadmin`**. En utilisant les autorisations du Compte A, connectez-vous à la console IAM et procédez comme suit :

i. Créez un utilisateur **`AccountAadmin`** et notez les informations d'identification de sécurité de l'utilisateur.

Pour obtenir des instructions, reportez-vous à la section Création d'un utilisateur IAM Compte AWS dans votre guide de l'utilisateur IAM.

ii. Accordez des privilèges d'administrateur à AccountAadminen joignant une politique utilisateur donnant un accès complet.

Pour obtenir des instructions, veuillez consulter la section Gestion des stratégies IAM dans le Guide de l'utilisateur IAM.

iii. Notez l'URL de connexion de l'utilisateur IAM pour. AccountAadmin Vous devez utiliser cette URL pour vous connecter sur la AWS Management Console. Pour plus d'informations sur l'emplacement de l'URL de connexion, voir Se connecter en AWS Management Console tant qu'utilisateur IAM dans le guide de l'utilisateur IAM. Notez l'URL de chacun des comptes.

2. Configurez le AWS CLI ou le AWS Tools for Windows PowerShell. Assurez-vous d'enregistrer les informations d'identification de l'utilisateur administrateur comme suit :

- Si vous utilisez le AWS CLI, créez un profil dans le fichier de configuration. `AccountAadmin`
- Si vous utilisez le AWS Tools for Windows PowerShell, assurez-vous de stocker les informations d'identification de la session sous le nom`AccountAadmin`.

Pour obtenir des instructions, veuillez consulter Configuration des outils pour les procédures pas à pas.

Étape 1 : créer des ressources dans le compte A et accorder des autorisations

À l'aide des informations d'identification de l'utilisateur figurant `AccountAadmin` dans le compte A et de l'URL de connexion spéciale de l'utilisateur IAM, connectez-vous au AWS Management Console et procédez comme suit :

1. Création des ressources d'un bucket et d'un utilisateur IAM

   a. Dans la console Amazon S3, créez un compartiment. Notez l'endroit Région AWS dans lequel vous avez créé le bucket. Pour obtenir des instructions, veuillez consulter Créer un compartiment.

   b. Dans la console IAM, procédez comme suit :

      i. Créez un utilisateur nommé Dave.

         Pour step-by-step obtenir des instructions, consultez la section Création d'utilisateurs IAM (console) dans le guide de l'utilisateur IAM.

      ii. Notez les `UserDave` informations d'identification.

      iii. Notez le nom de ressource Amazon (ARN) de l'utilisateur Dave. Dans la console IAM, sélectionnez l'utilisateur, et l'onglet Résumé fournit l'ARN de l'utilisateur.

2. Accordez des autorisations.

   Étant donné que le propriétaire du compartiment et le compte parent auquel appartient l'utilisateur sont identiques, ils Compte AWS peuvent accorder des autorisations à l'utilisateur en utilisant une politique de compartiment, une politique utilisateur ou les deux. Dans cet exemple, vous faites les deux. Si l'objet est également détenu par le même compte, le propriétaire du compartiment peut accorder des autorisations d'objet dans la stratégie de compartiment (ou une stratégie IAM).

   a. Dans la console Amazon S3, attachez la stratégie de compartiment suivante au compartiment *`awsexamplebucket1`*.

      La stratégie possède deux énoncés.

      - Le premier énoncé accorde à Dave les autorisations d'opération sur le compartiment `s3:GetBucketLocation` et `s3:ListBucket`.
      - Le second énoncé accorde l'autorisation `s3:GetObject`. Etant donné que le Compte A détient l'objet, l'administrateur du compte peut accorder l'autorisation `s3:GetObject`.

      Dans l'énoncé `Principal`, Dave est identifié par son ARN utilisateur. Pour en savoir plus sur les éléments de la stratégie, consultez Politiques et autorisations dans Amazon S3.

```
{
   "Version": "2012-10-17",
```

```
   "Statement": [
      {
         "Sid": "statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1"
         ]
      },
      {
         "Sid": "statement2",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
             "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

b. Créez une stratégie intégrée pour l'utilisateur Dave à l'aide de la stratégie suivante. La stratégie accorde à Dave l'autorisation `s3:PutObject`. Vous devez mettre à jour la stratégie en fournissant le nom du compartiment.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PermissionForObjectOperations",
         "Effect": "Allow",
         "Action": [
            "s3:PutObject"
```

```
            ],
            "Resource": [
               "arn:aws:s3:::awsexamplebucket1/*"
            ]
         }
      ]
}
```

Pour obtenir des instructions, reportez-vous à la section Gestion des politiques IAM dans le guide de l'utilisateur d'IAM. Notez que vous devez vous connecter sur la console grâce aux autorisations du Compte A.

Étape 2 : testez les autorisations

À l'aide des autorisations de Dave, vérifiez que les autorisations fonctionnent. Vous pouvez utiliser l'une des deux procédures suivantes.

Testez les autorisations à l'aide du AWS CLI

1. Mettez à jour le fichier de AWS CLI configuration en ajoutant le `UserDaveAccountA` profil suivant. Pour plus d'informations, consultez Configuration des outils pour les procédures pas à pas.

   ```
   [profile UserDaveAccountA]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Vérifiez que Dave peut exécuter les opérations telles qu'elles sont accordées dans la stratégie d'utilisateur. Téléchargez un exemple d'objet à l'aide de la AWS CLI `put-object` commande suivante.

   Le paramètre `--body` de la commande identifie le fichier source à charger. Par exemple, si le fichier se trouve à la racine du lecteur C : d'une Windows machine, vous devez le spécifier`c:\HappyFace.jpg`. Le paramètre `--key` fournit le nom de clé de l'objet.

   ```
   aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --
   body HappyFace.jpg --profile UserDaveAccountA
   ```

   Exécutez la AWS CLI commande suivante pour obtenir l'objet.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg
 --profile UserDaveAccountA
```

Testez les autorisations à l'aide du AWS Tools for Windows PowerShell

1. Conservez les informations d'identification de Dave sous le nom`AccountADave`. Vous utilisez ensuite ces informations d'identification pour PUT ajouter GET un objet.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountADave
   ```

2. Téléchargez un exemple d'objet à l'aide de la AWS Tools for Windows PowerShell `Write-S3Object` commande en utilisant les informations d'identification enregistrées par l'utilisateur Dave.

   ```
   Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg
    -StoredCredentials AccountADave
   ```

   Téléchargez l'objet précédemment chargé.

   ```
   Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -
   StoredCredentials AccountADave
   ```

Exemple 2 : propriétaire d'un compartiment accordant à ses utilisateurs des autorisations entre comptes sur un compartiment

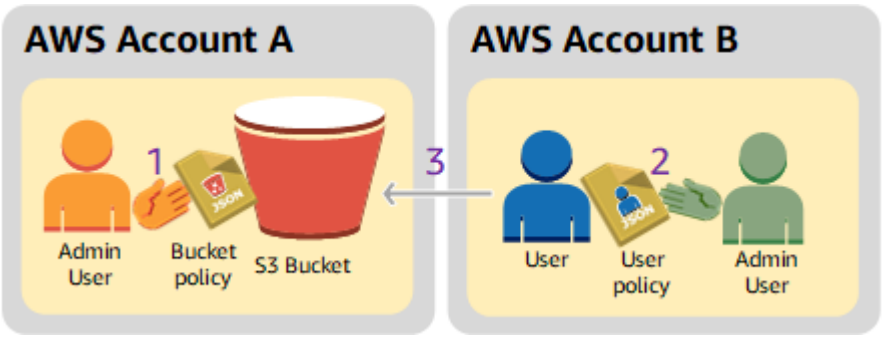
> **Important**
>
> Il est préférable d'accorder des autorisations à des rôles IAM plutôt qu'à des utilisateurs individuels. Pour savoir comment procéder, veuillez consulter Comprendre les autorisations entre comptes et utiliser les rôles IAM.

Rubriques

Un compte Compte AWS A, par exemple, peut accorder à un autre Compte AWS, le compte B, l'autorisation d'accéder à ses ressources, telles que des seaux et des objets. Le compte B peut ensuite déléguer ces autorisations aux utilisateurs du compte. Dans ce scénario fictif, le propriétaire d'un compartiment accorde une autorisation entre comptes à un autre compte pour lui permettre d'effectuer des opérations spécifiques sur un compartiment.

> **Note**
>
> Le compte A peut également accorder directement des autorisations à un utilisateur du compte B, par le biais d'une stratégie de compartiment. Cependant, l'utilisateur aura toujours besoin de l'autorisation du compte parent, le compte B, auquel il appartient, même si le compte B n'a pas les autorisations du compte A. Tant que l'utilisateur dispose de l'autorisation du propriétaire de la ressource et du compte parent, il pourra accéder à la ressource.

Les étapes suivantes résument la procédure à suivre :



1. L'utilisateur administrateur du compte A attache une stratégie de compartiment en accordant au compte B des autorisations inter-comptes pour lui permettre d'effectuer des opérations spécifiques sur un compartiment.

   Notez que l'utilisateur administrateur du compte B hérite automatiquement des autorisations.
2. L'utilisateur administrateur du compte B attache une stratégie d'utilisateur à l'utilisateur en déléguant les autorisations qu'il a reçues du compte A.
3. L'utilisateur du compte B vérifie ensuite les autorisations en accédant à un objet dans le compartiment appartenant au compte A.

Pour cet exemple, vous avez besoin de deux comptes. Le tableau ci-dessous montre comment nous faisons référence à ces comptes et à leurs utilisateurs administrateurs. Conformément aux directives IAM (voirUtilisation d'un utilisateur administrateur pour créer des ressources et accorder des autorisations), nous n'utilisons pas les informations d'identification de l'utilisateur root dans cette procédure pas à pas. A la place, vous créez un utilisateur administrateur dans chaque compte et utilisez ces autorisations pour créer des ressources et leur accorder des autorisations.

Compte AWS ID	Compte désigné comme	Utilisateur administrateur du compte
1111-1111-1111	Compte A	AccountAadmin
2222-2222-2222	Compte B	AccountBadmin

Toutes les tâches de création d'utilisateurs et d'octroi d'autorisations sont effectuées dans la AWS Management Console. Pour vérifier les autorisations, la procédure pas à pas utilise les outils de ligne de commande AWS Command Line Interface (CLI) et AWS Tools for Windows PowerShell vous n'avez donc pas besoin d'écrire de code.

Préparation de la procédure détaillée

1. Assurez-vous que vous en avez deux Comptes AWS et que chaque compte possède un utilisateur administrateur, comme indiqué dans le tableau de la section précédente.

   a. Inscrivez-vous pour un Compte AWS, si nécessaire.

   b. Utilisez les autorisations du compte A pour vous connecter à la console IAM afin de créer l'utilisateur administrateur :

      i. Créez un utilisateur **`AccountAadmin`** et notez les informations d'identification de sécurité. Pour obtenir des instructions, veuillez consulter la section Création d'un utilisateur IAM dans votre Compte AWS du Guide de l'utilisateur IAM.

      ii. Accordez des privilèges d'administrateur à AccountAadminen joignant une politique utilisateur donnant un accès complet. Pour obtenir des instructions, veuillez consulter Utilisation de stratégies dans le Guide de l'utilisateur IAM.

   c. Lorsque vous êtes dans la console IAM, notez l'URL de connexion de l'utilisateur IAM sur le tableau de bord. Tous les utilisateurs du compte doivent utiliser cette URL pour se connecter à la AWS Management Console.

      Pour plus d’informations, consultez Comment les utilisateurs se connectent à votre compte dans le Guide de l’utilisateur IAM.

   d. Répétez l'étape précédente en utilisant les informations d'identification du compte B et créez un utilisateur administrateur**`AccountBadmin`**.

2. Configurez le AWS Command Line Interface (AWS CLI) ou le AWS Tools for Windows PowerShell. Assurez-vous d'enregistrer les informations d'identification de l'utilisateur administrateur comme suit :

   - Si vous utilisez le AWS CLI, créez deux profils, `AccountAadmin` et`AccountBadmin`, dans le fichier de configuration.
   - Si vous utilisez le AWS Tools for Windows PowerShell, assurez-vous de stocker les informations d'identification de la session sous forme `AccountAadmin` et`AccountBadmin`.

   Pour obtenir des instructions, veuillez consulter Configuration des outils pour les procédures pas à pas.

3. Enregistrez les autorisations de l'utilisateur administrateur, également appelées « profils ». Vous pouvez utiliser le nom de profil au lieu de spécifier des autorisations pour chaque commande

saisie. Pour plus d'informations, consultez Configuration des outils pour les procédures pas à pas.

a. Ajoutez des profils dans le fichier AWS CLI d'informations d'identification pour chacun des utilisateurs administrateurs `AccountAadmin` et `AccountBadmin` dans les deux comptes.

```
[AccountAadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1
```

b. Si vous utilisez le AWS Tools for Windows PowerShell, exécutez la commande suivante.

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-access-key -storeas AccountAadmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-access-key -storeas AccountBadmin
```

Étape 1 : Tâches du compte A

Étape 1.1 : Connectez-vous au AWS Management Console

À l'aide de l'URL de connexion de l'utilisateur IAM pour le compte A, connectez-vous d'abord au compte en AWS Management Console tant qu'AccountAadminutilisateur. Cet utilisateur créera un compartiment et y attachera une stratégie.

Étape 1.2 : Créer un compartiment

1. Dans la console Amazon S3, créez un compartiment. Cet exercice suppose que le bucket a été créé dans l'est des États-Unis (Virginie du Nord) Région AWS et qu'il porte un nom*DOC-EXAMPLE-BUCKET*.

   Pour obtenir des instructions, veuillez consulter Créer un compartiment.
2. Chargez un exemple d'objet dans le compartiment.

   Pour plus d'informations, consultez Étape 2 : Charger un objet dans votre compartiment.

Étape 1.3 : attacher une stratégie de compartiment afin d'accorder des autorisations entre comptes au compte B

La politique du compartiment accorde les `s3:ListBucket` autorisations `s3:GetLifecycleConfiguration` et au compte B. On suppose que vous êtes toujours connecté à la console à l'aide des informations AccountAadmind'identification utilisateur.

1. Attachez la stratégie de compartiment suivante à *DOC-EXAMPLE-BUCKET*. Cette stratégie accorde au compte B une autorisation pour les actions `s3:GetLifecycleConfiguration` et `s3:ListBucket`.

   Pour obtenir des instructions, veuillez consulter Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:GetLifecycleConfiguration",
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
         }
      ]
   }
   ```

2. Vérifiez que le compte B (et donc son utilisateur administrateur) peut effectuer les opérations.

   - Vérifiez à l'aide du AWS CLI

     ```
     aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
     aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --
     profile AccountBadmin
     ```

- Vérifiez à l'aide du AWS Tools for Windows PowerShell

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -
StoredCredentials AccountBadmin
```

Étape 2 : Tâches du compte B

L'administrateur du compte B va maintenant créer un utilisateur, Dave, et lui déléguer les autorisations reçues du compte A.

Étape 2.1 : Connectez-vous au AWS Management Console

À l'aide de l'URL de connexion utilisateur IAM pour le compte B, connectez-vous d'abord au compte en AWS Management Console tant qu'AccountBadminutilisateur.

Étape 2.2 : créer l'utilisateur Dave dans le compte B

Dans la console IAM, créez un utilisateur,**`Dave`**.

Pour obtenir des instructions, veuillez consulter la section Création d'utilisateurs IAM (console) du Guide de l'utilisateur IAM

Étape 2.3 : déléguer les autorisations à l'utilisateur Dave

Créez une stratégie intégrée pour l'utilisateur Dave à l'aide de la stratégie suivante. Vous devrez mettre à jour la stratégie en fournissant le nom du compartiment.

Il est supposé que vous êtes connecté à la console à l'aide des informations AccountBadmind'identification utilisateur.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example",
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
```

```
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
         ]
      }
   ]
}
```

Pour obtenir des instructions, veuillez consulter la section Gestion des stratégies IAM dans le Guide de l'utilisateur IAM.

Étape 2.4 : Testez les autorisations

Dans le compte B, Dave peut maintenant répertorier le contenu du compartiment *DOC-EXAMPLE-BUCKET* appartenant au compte A. Vous pouvez vérifier les autorisations via l'une des procédures suivantes.

Testez les autorisations à l'aide du AWS CLI

1. Ajoutez le `UserDave` profil au fichier de AWS CLI configuration. Pour plus d'informations sur le fichier de configuration, consultez Configuration des outils pour les procédures pas à pas.

   ```
   [profile UserDave]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. À l'invite de commande, entrez la AWS CLI commande suivante pour vérifier que Dave peut désormais obtenir une liste d'objets à partir du compte *DOC-EXAMPLE-BUCKET* appartenant au compte A. Notez que la commande spécifie le `UserDave` profil.

   ```
   aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile UserDave
   ```

   Dave n'a aucune autre autorisation. Ainsi, s'il tente une autre opération, par exemple la `get-bucket-lifecycle` configuration suivante, Amazon S3 renvoie l'autorisation refusée.

   ```
   aws s3api get-bucket-lifecycle-configuration --bucket DOC-EXAMPLE-BUCKET --profile
    UserDave
   ```

Tester les autorisations en utilisant AWS Tools for Windows PowerShell

1. Conservez les informations d'identification de Dave sous le nom`AccountBDave`.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
 AccountBDave
```

2. Essayez la commande permettant de répertorier les compartiments.

```
get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
```

Dave n'a aucune autre autorisation. Ainsi, s'il tente une autre opération, par exemple la suivante, `get-s3bucketlifecycleconfiguration` Amazon S3 renvoie l'autorisation refusée.

```
get-s3bucketlifecycleconfiguration -BucketName DOC-EXAMPLE-BUCKET -
StoredCredentials AccountBDave
```

Étape 3 : (Facultatif) essayer le refus explicite

Vous pouvez obtenir des autorisations à l'aide d'une liste de contrôle d'accès (ACL), d'une politique de compartiment ou d'une politique utilisateur. Mais si un refus explicite est défini par une politique de compartiment ou une politique utilisateur, le refus explicite a priorité sur toute autre autorisation. Pour les tests, mettez à jour la politique du compartiment et refusez explicitement l'`s3:ListBucket`autorisation au compte B. La politique accorde également une `s3:ListBucket` autorisation. Toutefois, le refus explicite a priorité, et le compte B ou les utilisateurs du compte B ne pourront pas y répertorier les *DOC-EXAMPLE-BUCKET* objets.

1. À l'aide des informations d'identification de l'utilisateur `AccountAadmin` dans le compte A, remplacez la politique de compartiment par la suivante.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example permissions",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:GetLifecycleConfiguration",
            "s3:ListBucket"
         ],
```

```
            "Resource": [
               "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
         },
         {
            "Sid": "Deny permission",
            "Effect": "Deny",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
         }
      ]
}
```

2. Désormais, si vous essayez d'obtenir une liste de choses à faire à l'aide `AccountBadmin` d'informations d'identification, l'accès est refusé.

   - À l'aide de AWS CLI, exécutez la commande suivante :

     ```
     aws s3 ls s3://DOC-EXAMPLE-BUCKET --profile AccountBadmin
     ```

   - À l'aide de AWS Tools for Windows PowerShell, exécutez la commande suivante :

     ```
     get-s3object -BucketName DOC-EXAMPLE-BUCKET -StoredCredentials AccountBDave
     ```

Étape 4 : Nettoyer

1. Une fois les tests terminés, vous pouvez effectuer les opérations suivantes pour effectuer le nettoyage :

   - Connectez-vous au AWS Management Console (AWS Management Console) à l'aide des informations d'identification du compte A, puis procédez comme suit :

     - Dans la console Amazon S3, supprimez la stratégie de compartiment attachée à examplebucket. Dans la page Properties du compartiment, supprimez la stratégie dans la section Permissions.

- Si le compartiment a été créé pour cet exercice, supprimez les objets, puis le compartiment, dans la console Amazon S3.
- Dans la console IAM, supprimez l'`AccountAadmin`utilisateur.

2. Connectez-vous à la console IAM à l'aide des informations d'identification du compte B. Supprimer l'utilisateur`AccountBadmin`. Pour step-by-step obtenir des instructions, reportez-vous à la section Suppression d'un utilisateur IAM dans le guide de l'utilisateur IAM.

Exemple 3 : propriétaire d'un compartiment accordant des autorisations sur des objets qu'il ne possède pas

> **Important**
>
> Il est préférable d'accorder des autorisations à des rôles IAM plutôt qu'à des utilisateurs individuels. Pour savoir comment procéder, veuillez consulter Comprendre les autorisations entre comptes et utiliser les rôles IAM.

Rubriques

- Étape 0 : Préparez-vous à suivre la procédure
- Étape 1 : Tâches du compte A
- Étape 2 : Tâches du compte B
- Étape 3 : Tester les autorisations
- Étape 4 : Nettoyer

Dans cet exemple, le scénario est le suivant : le propriétaire du compartiment souhaite autoriser l'accès aux objets, mais le propriétaire du compartiment ne possède pas tous les objets du compartiment. Pour cet exemple, le propriétaire du compartiment essaie d'accorder une autorisation à des utilisateurs de son propre compte.

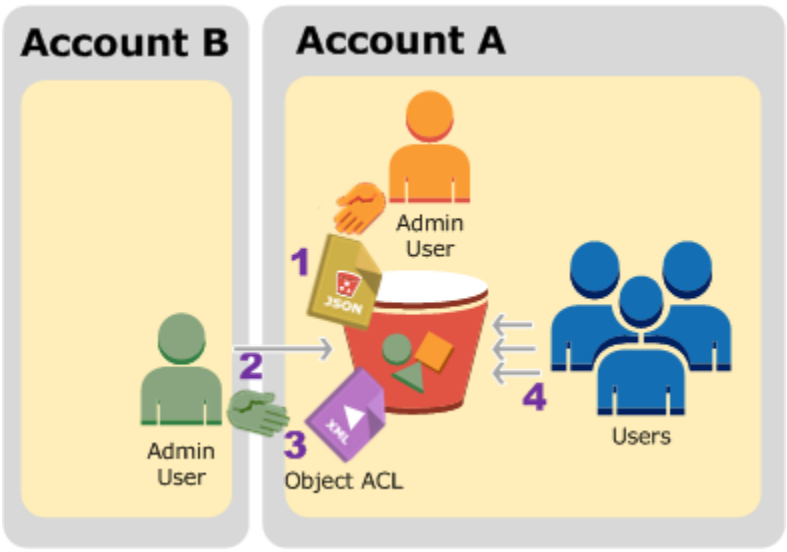
Le propriétaire d'un bucket peut autoriser d'autres Comptes AWS personnes à télécharger des objets. Par défaut, le propriétaire du compartiment ne possède pas d'objet écrit dans un compartiment par un autre Compte AWS. Les objets sont la propriété des comptes qui les écrivent dans un compartiment S3. Si le propriétaire du compartiment ne possède aucun objet dans le compartiment, il doit d'abord accorder l'autorisation au propriétaire du compartiment à l'aide d'une liste de contrôle d'accès aux objets (ACL). Le propriétaire du compartiment peut ensuite accorder des autorisations à un objet dont

il n'est pas le propriétaire. Pour plus d'informations, consultez Propriété du compartiment et de l'objet Amazon S3.

Si le propriétaire du compartiment applique le paramètre appliqué par le propriétaire du compartiment pour la propriété de l'objet S3 pour le compartiment, le propriétaire du compartiment possèdera tous les objets du compartiment, y compris les objets écrits par un autre Compte AWS. Cette approche résout le problème selon lequel les objets ne sont pas la propriété du propriétaire du compartiment. Vous pouvez ensuite déléguer des autorisations à des utilisateurs de votre propre compte ou à d'autres Comptes AWS.

> **Note**
>
> La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.
>
> La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Dans cet exemple, nous supposons que le propriétaire du compartiment n'a pas appliqué le paramètre appliqué par le propriétaire du compartiment pour la propriété de l'objet. Le propriétaire du compartiment délègue l'autorisation aux utilisateurs figurant dans son propre compte. Voici un résumé des étapes de la procédure pas à pas :



1. L'utilisateur administrateur du compte A attache une stratégie de compartiment contenant deux instructions.
   - Accorder une autorisation entre comptes au compte B pour charger des objets.
   - Autoriser un utilisateur dans son propre compte à accéder aux objets figurant dans le compartiment.
2. L'utilisateur administrateur du compte B charge des objets dans le compartiment appartenant au compte A.
3. L'administrateur du compte B met à jour la liste ACL d'objets en ajoutant une affectation qui donne au propriétaire du compartiment l'autorisation de contrôle total sur l'objet.
4. L'utilisateur figurant dans le compte A vérifie cela en accédant aux objets du compartiment, quel que soit leur propriétaire.

Pour cet exemple, vous avez besoin de deux comptes. Le tableau ci-dessous montre comment nous faisons référence à ces comptes et aux utilisateurs administrateurs dans ces comptes. Dans cette démonstration, vous n'utiliserez pas les autorisations d'utilisateur root du compte, conformément aux directives IAM recommandées. Pour plus d'informations, consultez Utilisation d'un utilisateur administrateur pour créer des ressources et accorder des autorisations. À la place, vous allez créer un administrateur dans chaque compte et utiliser ces autorisations pour créer des ressources et leur accorder des autorisations.

Compte AWS ID	Compte désigné comme	Administrateur du compte
1111-1111-1111	Compte A	AccountAadmin
2222-2222-2222	Compte B	AccountBadmin

Toutes les tâches de création d'utilisateurs et d'octroi d'autorisations sont effectuées dans la AWS Management Console. Pour vérifier les autorisations, la procédure pas à pas utilise les outils de ligne de commande AWS Command Line Interface (AWS CLI) et AWS Tools for Windows PowerShell vous n'avez donc pas besoin d'écrire de code.

Étape 0 : Préparez-vous à suivre la procédure

1. Assurez-vous d'en avoir deux Comptes AWS et d'avoir un administrateur pour chaque compte, comme indiqué dans le tableau de la section précédente.

   a. Inscrivez-vous pour un Compte AWS, si nécessaire.

   b. À l'aide des informations d'identification du compte A, connectez-vous à la console IAM et procédez comme suit pour créer un utilisateur administrateur :

      - Créez un utilisateur **`AccountAadmin`** et notez les informations de sécurité de l'utilisateur. Pour plus d'informations sur l'ajout d'utilisateurs, consultez la section Création d'un utilisateur IAM dans votre Compte AWS du Guide de l'utilisateur IAM.
      - Accordez des autorisations d'administrateur à AccountAadminen joignant une politique utilisateur qui donne un accès complet. Pour obtenir des instructions, veuillez consulter la section Gestion des stratégies IAM dans le Guide de l'utilisateur IAM.
      - Dans le tableau de bord de la console IAM, notez l'URL de connexion de l'utilisateur IAM. Les utilisateurs figurant dans ce compte doivent utiliser cette URL pour se connecter à la AWS Management Console. Pour plus d'informations, consultez Comment les utilisateurs se connectent à votre compte dans le Guide de l'utilisateur IAM.

   c. Répétez l'étape précédente en utilisant les informations d'identification du compte B et créez un utilisateur administrateur**`AccountBadmin`**.

2. Configurez le AWS CLI ou les outils pour Windows PowerShell. Veillez à enregistrer les autorisations de l'administrateur comme suit :

   - Si vous utilisez le AWS CLI, créez deux profils, `AccountAadmin` et`AccountBadmin`, dans le fichier de configuration.

- Si vous utilisez les Outils pour Windows PowerShell, assurez-vous de stocker les informations d'identification de la session sous forme `AccountAadmin` et`AccountBadmin`.

Pour obtenir des instructions, veuillez consulter Configuration des outils pour les procédures pas à pas.

Étape 1 : Tâches du compte A

Effectuez les opérations suivantes pour le compte A :

Étape 1.1 : Se connecter à la console

À l'aide de l'URL de connexion de l'utilisateur IAM pour le compte A, connectez-vous au compte en AWS Management Console tant qu'**`AccountAadmin`**utilisateur. Cet utilisateur créera un compartiment et y attachera une stratégie.

Étape 1.2 : Créer un compartiment et un utilisateur, puis ajouter une stratégie de compartiment accordant des autorisations à l'utilisateur

1. Dans la console Amazon S3, créez un compartiment. Cet exercice suppose que le bucket a été créé dans l'est des États-Unis (Virginie du Nord) Région AWS et que son nom est*`example-s3-bucket1`*.

   Pour obtenir des instructions, veuillez consulter Créer un compartiment.
2. Dans la console IAM, créez un utilisateur**`Dave`**.

   Pour step-by-step obtenir des instructions, consultez la section Création d'utilisateurs IAM (console) dans le guide de l'utilisateur IAM.
3. Notez les informations d'identification de l'utilisateur Dave.
4. Dans la console Amazon S3, attachez la stratégie de compartiment suivante au compartiment *`example-s3-bucket1`*. Pour obtenir des instructions, veuillez consulter Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3. Suivez les étapes pour ajouter une stratégie de compartiment. Pour plus d'informations sur la façon de trouver les identifiants de compte, consultez la section Trouver votre Compte AWS identifiant.

   Cette stratégie accorde au compte B les autorisations `s3:PutObject` et `s3:ListBucket`. La politique accorde également l'`s3:GetObject`autorisation `Dave` à l'utilisateur.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:PutObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::example-s3-bucket1/*",
            "arn:aws:s3:::example-s3-bucket1"
         ]
      },
      {
         "Sid": "Statement3",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::example-s3-bucket1/*"
         ]
      }
   ]
}
```

Étape 2 : Tâches du compte B

Maintenant que le compte B est autorisé à effectuer des opérations sur le compartiment du compte A, l'administrateur du compte B effectue les opérations suivantes :

- Télécharge un objet dans le compartiment du compte A

- Ajoute une autorisation dans l'ACL de l'objet pour permettre au compte A, le propriétaire du compartiment, de contrôler totalement

En utilisant le AWS CLI

1. À l'aide de la `put-object` AWS CLI commande, chargez un objet. Le paramètre `--body` de la commande identifie le fichier source à charger. Par exemple, si le fichier se trouve sur le `C:` lecteur d'une Windows machine, spécifiez`c:\HappyFace.jpg`. Le paramètre `--key` fournit le nom de clé de l'objet.

```
aws s3api put-object --bucket example-s3-bucket1 --key HappyFace.jpg --body
 HappyFace.jpg --profile AccountBadmin
```

2. Ajoutez une affectation dans la liste ACL d'objets pour accorder au propriétaire du compartiment le contrôle total de l'objet. Pour plus d'informations sur la façon de trouver un nom d'utilisateur canonique, voir Trouver l'identifiant d'utilisateur canonique correspondant à votre nom Compte AWS dans le Guide de référence de gestion de AWS compte.

```
aws s3api put-object-acl --bucket example-s3-bucket1 --key HappyFace.jpg --grant-
full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Utilisation des outils pour Windows PowerShell

1. À l'aide de la `Write-S3Object` commande, chargez un objet.

```
Write-S3Object -BucketName example-s3-bucket1 -key HappyFace.jpg -file
 HappyFace.jpg -StoredCredentials AccountBadmin
```

2. Ajoutez une affectation dans la liste ACL d'objets pour accorder au propriétaire du compartiment le contrôle total de l'objet.

```
Set-S3ACL -BucketName example-s3-bucket1 -Key HappyFace.jpg -CannedACLName "bucket-
owner-full-control" -StoredCreden
```

Étape 3 : Tester les autorisations

À présent, vérifiez que l'utilisateur Dave du compte A peut accéder à l'objet appartenant au compte B.

En utilisant le AWS CLI

1. Ajoutez les informations d'identification de l'utilisateur Dave au fichier de AWS CLI configuration et créez un nouveau profil,`UserDaveAccountA`. Pour plus d'informations, consultez Configuration des outils pour les procédures pas à pas.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Exécutez la commande CLI `get-object` pour télécharger `HappyFace.jpg`, puis enregistrez le fichier localement. Vous attribuez à l'utilisateur Dave des informations d'identification en ajoutant le paramètre `--profile`.

```
aws s3api get-object --bucket example-s3-bucket1 --key HappyFace.jpg Outputfile.jpg
 --profile UserDaveAccountA
```

Utilisation des outils pour Windows PowerShell

1. Stockez les AWS informations d'identification de l'utilisateur Dave`UserDaveAccountA`, comme dans le magasin permanent.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-
SecretAccessKey -storeas UserDaveAccountA
```

2. Exécutez la commande `Read-S3Object` pour télécharger `HappyFace.jpg`, puis enregistrez le fichier localement. Vous attribuez à l'utilisateur Dave des informations d'identification en ajoutant le paramètre `-StoredCredentials`.

```
Read-S3Object -BucketName example-s3-bucket1 -Key HappyFace.jpg -file HappyFace.jpg
  -StoredCredentials UserDaveAccountA
```

Étape 4 : Nettoyer

1. Une fois les tests terminés, vous pouvez effectuer les opérations suivantes pour effectuer le nettoyage :

- Connectez-vous à la AWS Management Console en utilisant les autorisations du compte A, et procédez comme suit :
  - Dans la console Amazon S3, supprimez la politique de compartiment attachée à *example-s3-bucket1*. Dans la page Properties du compartiment, supprimez la stratégie dans la section Permissions.
  - Si le compartiment a été créé pour cet exercice, supprimez les objets, puis le compartiment, dans la console Amazon S3.
  - Dans la console IAM, supprimez l'AccountAadminutilisateur. Pour step-by-step obtenir des instructions, reportez-vous à la section Suppression d'un utilisateur IAM dans le guide de l'utilisateur IAM.

2. Connectez-vous à la AWS Management Console en utilisant les autorisations du compte B. Dans la console IAM, supprimez l'utilisateur AccountBadmin.

Exemple 4 - Le propriétaire du bucket accorde une autorisation multicompte à des objets qu'il ne possède pas

Rubriques

Dans cet exemple de scénario, vous êtes propriétaire d'un bucket et vous avez autorisé d'autres utilisateurs Comptes AWS à télécharger des objets. Si vous avez appliqué le paramètre appliqué par le propriétaire du compartiment pour la propriété de l'objet S3 pour le compartiment, vous posséderez tous les objets du compartiment, y compris les objets écrits par un autre Compte AWS. Cette approche résout le problème selon lequel les objets ne vous appartiennent pas, en tant que propriétaire du compartiment. Vous pouvez ensuite déléguer des autorisations à des utilisateurs de votre propre compte ou à d'autres Comptes AWS. Supposons que le paramètre appliqué par

le propriétaire du compartiment pour la propriété de l'objet S3 ne soit pas activé. C'est pourquoi le compartiment peut avoir des objets détenus par d'autres Comptes AWS .

A présent, admettons qu'en tant que propriétaire du compartiment, vous devez accorder des autorisations entre comptes sur des objets, quel que soit le propriétaire, à un utilisateur d'un autre compte. Par exemple, cet utilisateur peut être une application de facturation qui a besoin d'accéder aux métadonnées d'objet. Deux problèmes majeurs se posent :

- Le propriétaire du compartiment n'a aucune autorisation sur ces objets créés par d'autres Comptes AWS. Pour que le propriétaire du bucket puisse accorder des autorisations sur des objets qui ne lui appartiennent pas, il doit d'abord accorder l'autorisation au propriétaire du bucket. Le propriétaire de l'objet est celui Compte AWS qui a créé les objets. Le propriétaire du compartiment peut alors déléguer ces autorisations.
- Le compte propriétaire du bucket peut déléguer des autorisations aux utilisateurs de son propre compte (voirExemple 3 : propriétaire d'un compartiment accordant des autorisations sur des objets qu'il ne possède pas). Toutefois, le compte du propriétaire du compartiment ne peut pas déléguer d'autorisations à d'autres personnes, Comptes AWS car la délégation entre comptes n'est pas prise en charge.

Dans ce scénario, le propriétaire du compartiment peut créer un rôle AWS Identity and Access Management (IAM) autorisé à accéder aux objets. Ensuite, le propriétaire du compartiment peut accorder une autre Compte AWS autorisation pour assumer le rôle, lui permettant ainsi temporairement d'accéder aux objets du compartiment.

> **Note**
>
> La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.
> La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques

pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Comprendre les autorisations entre comptes et utiliser les rôles IAM

Les rôles IAM permettent à plusieurs scénarios de déléguer l'accès aux ressources, et l'accès entre comptes est l'un des scénarios clés. Dans cet exemple, le propriétaire du compartiment, le compte A, utilise un rôle IAM pour déléguer temporairement l'accès aux objets entre comptes aux utilisateurs d'un autre Compte AWS, le compte C. Chaque rôle IAM que vous créez est associé aux deux politiques suivantes :

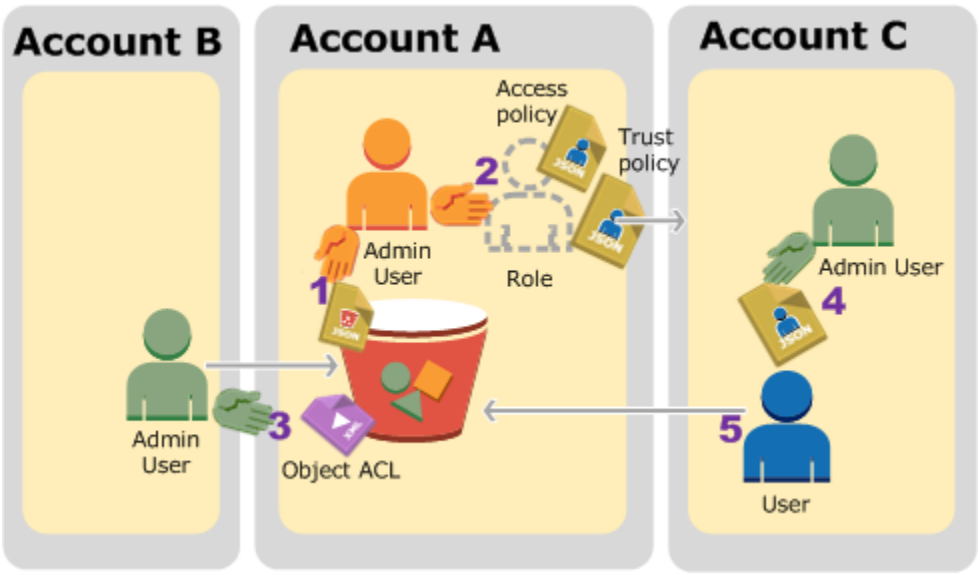
- Une politique de confiance identifiant une autre Compte AWS personne qui peut assumer le rôle.
- Une stratégie d'accès qui définit les autorisations : par exemple, `s3:GetObject`, accordées lorsqu'une personne assume le rôle. Pour obtenir une liste d'autorisations que vous pouvez spécifier dans une stratégie, consultez Actions politiques pour Amazon S3.

La Compte AWS personne identifiée dans la politique de confiance accorde ensuite à son utilisateur l'autorisation d'assumer le rôle. L'utilisateur peut ensuite procéder comme suit pour accéder aux objets :

- Assumez le rôle et, en réponse, obtenir des autorisations de sécurité temporaires.
- Accédez aux objets dans le compartiment grâce aux autorisations de sécurité temporaires.

Pour plus d'informations sur les rôles IAM, consultez Rôles IAM dans le Guide de l'utilisateur IAM.

Voici un résumé des étapes de la procédure pas à pas :



1. L'utilisateur administrateur du Compte A attache une stratégie de compartiment en accordant au Compte B l'autorisation conditionnelle de charger des objets.
2. L'administrateur du Compte A crée un rôle IAM, en instaurant la confiance avec le Compte C, afin que les utilisateurs de ce compte puissent accéder au Compte A. La stratégie d'accès attachée au rôle limite les actions de l'utilisateur du Compte C lorsque ce dernier accède au Compte A.
3. L'administrateur du compte B charge un objet dans le compartiment détenu par le Compte A, en accordant l'autorisation de contrôle total au propriétaire du compartiment.
4. L'administrateur du Compte C crée un utilisateur et attache une stratégie d'utilisateur qui permet à l'utilisateur d'assumer le rôle.
5. L'utilisateur du Compte C assume tout d'abord le rôle, qui lui renvoie les autorisations de sécurité temporaires. Grâce à ces autorisations, l'utilisateur accède ensuite aux objets du compartiment.

Pour cet exemple, vous avez besoin de trois comptes. Le tableau ci-dessous montre comment nous faisons référence à ces comptes et aux utilisateurs administrateurs dans ces comptes. Conformément aux directives de l'IAM (voirUtilisation d'un utilisateur administrateur pour créer des ressources et accorder des autorisations), nous n'utilisons pas les Utilisateur racine d'un compte AWS informations d'identification dans cette procédure pas à pas. A la place, vous créez un utilisateur administrateur dans chaque compte et utilisez ces autorisations pour créer des ressources et leur accorder des autorisations.

Compte AWS ID	Compte désigné comme	Utilisateur administrateur du compte
1111-1111-1111	Compte A	AccountAadmin
2222-2222-2222	Compte B	AccountBadmin
3333-3333-3333	Compte C	AccountCadmin

Étape 0 : Préparez-vous à suivre la procédure

> **Note**
>
> Vous souhaiterez peut-être ouvrir un éditeur de texte et noter certaines informations au fur et à mesure que vous suivez les étapes. En particulier, vous avez besoin d'ID de compte, d'ID d'utilisateur canonique, d'URL de connexion d'utilisateur IAM pour chaque compte pour vous connecter à la console, et des Amazon Resource Names (ARN) des utilisateurs IAM et de rôles.

1. Assurez-vous que vous en avez trois Comptes AWS et que chaque compte possède un utilisateur administrateur, comme indiqué dans le tableau de la section précédente.

   a. Inscrivez-vous Comptes AWS, au besoin. Nous appelons ces comptes : Compte A, Compte B et Compte C.

   b. Utilisez les autorisations du compte A pour vous connecter à la IAM console (Console IAM) et procédez comme suit pour créer un utilisateur administrateur :

      - Créez un utilisateur **`AccountAadmin`** et notez ses informations de sécurité. Pour plus d'informations sur l'ajout d'utilisateurs, consultez la section Création d'un utilisateur IAM dans votre Compte AWS du Guide de l'utilisateur IAM.
      - Accordez des privilèges d'administrateur à AccountAadminen joignant une politique utilisateur donnant un accès complet. Pour obtenir des instructions, veuillez consulter la section Gestion des stratégies IAM dans le Guide de l'utilisateur IAM.
      - Dans le tableau de bord de la console IAM, notez l'URL de connexion de l'utilisateur IAM. Les utilisateurs figurant dans ce compte doivent utiliser cette URL pour se connecter

à la AWS Management Console. Pour plus d'informations, voir Se connecter en AWS Management Console tant qu'utilisateur IAM dans le guide de l'utilisateur IAM.

c. Répétez l'étape précédente pour créer les utilisateurs administrateur dans le Compte B et le Compte C.

2. Pour le compte C, notez l'ID utilisateur canonique.

   Lorsque vous créez un rôle IAM dans le Compte A, la stratégie d'approbation accorde au Compte C l'autorisation d'assumer le rôle en spécifiant l'ID de compte. Vous pouvez trouver les informations de compte comme suit :

   a. Utilisez votre Compte AWS identifiant ou alias de compte, votre nom d'utilisateur IAM et votre mot de passe pour vous connecter à la console Amazon S3.

   b. Choisissez le nom d'un compartiment Amazon S3 pour afficher les détails concernant ce compartiment.

   c. Sélectionnez l'onglet Permissions (Autorisations), puis Access Control List (Liste de contrôle d'accès).

   d. Dans la section Accès à votre Compte AWS, dans la colonne Account (Compte) figure un identifiant long tel que `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Ceci est votre ID d'utilisateur canonique.

3. Lorsque vous créez une stratégie de compartiment, vous avez besoin des informations suivantes. Notez ces valeurs :

   - ID d'utilisateur canonique du Compte A – Lorsque l'administrateur du Compte A accorde une autorisation conditionnelle de téléchargement d'objet à l'administrateur du Compte B, la condition spécifie l'ID d'utilisateur canonique de l'utilisateur du Compte A qui doit obtenir le contrôle total des objets.

     > **Note**
     >
     > Le concept d'ID d'utilisateur canonique est propre à Amazon S3. Il s'agit d'une version cryptée de 64 caractères de l'ID de compte.

   - ARN de l'utilisateur pour l'administrateur du compte B : vous pouvez trouver l'ARN de l'utilisateur dans la console IAM. Vous devez sélectionner l'utilisateur et trouver son ARN dans l'onglet Résumé.

Dans la politique du bucket, vous `AccountBadmin` autorisez le téléchargement d'objets et vous spécifiez l'utilisateur à l'aide de l'ARN. Voici un exemple de valeur ARN :

```
arn:aws:iam::AccountB-ID:user/AccountBadmin
```

4. Configurez le AWS Command Line Interface (CLI) ou le AWS Tools for Windows PowerShell. Assurez-vous d'enregistrer les informations d'identification de l'utilisateur administrateur comme suit :

   - Si vous utilisez le AWS CLI, créez des profils`AccountBadmin`, `AccountAadmin` et dans le fichier de configuration.
   - Si vous utilisez le AWS Tools for Windows PowerShell, assurez-vous de stocker les informations d'identification de la session sous forme `AccountAadmin` et`AccountBadmin`.

   Pour obtenir des instructions, veuillez consulter Configuration des outils pour les procédures pas à pas.

Étape 1 : Réalisation des tâches pour le compte A

Dans cet exemple, le Compte A est le propriétaire du compartiment. L'utilisateur AccountAadmin du compte A effectuera donc ce qui suit :

- Créez un compartiment.
- Joignez une politique de compartiment qui accorde à l'administrateur du compte B l'autorisation de télécharger des objets.
- Créez un rôle IAM qui accorde au compte C l'autorisation d'assumer le rôle afin qu'il puisse accéder aux objets du compartiment.

Étape 1.1 : Connectez-vous au AWS Management Console

À l'aide de l'URL de connexion de l'utilisateur IAM pour le compte A, connectez-vous d'abord au compte en AWS Management Console tant qu'**`AccountAadmin`**utilisateur. Cet utilisateur créera un compartiment et y attachera une stratégie.

Étape 1.2 : créez un compartiment et attachez une stratégie de compartiment

Dans la console Amazon S3, procédez comme suit :

1. Créez un compartiment. Cet exercice assume que le nom du compartiment est *example-s3-bucket1*.

   Pour obtenir des instructions, veuillez consulter Créer un compartiment.

2. Joignez la politique de compartiment suivante. La politique accorde une autorisation conditionnelle à l'administrateur du compte B pour télécharger des objets.

   Mettez à jour la politique en fournissant vos propres valeurs pour *example-s3-bucket1AccountB-ID*, et le*CanonicalUserId-of-AWSaccountA-BucketOwner*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket1/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::example-s3-bucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
                }
            }
        }
    ]
}
```

Étape 1.3 : Création d'un rôle IAM pour autoriser l'accès croisé au compte C dans le compte A

Dans la console IAM, créez un rôle IAM (**`examplerole`**) qui accorde au compte C l'autorisation d'assumer le rôle. Assurez-vous que vous êtes toujours connecté en tant qu'administrateur du compte A, car le rôle doit être créé dans le compte A.

1. Avant de créer le rôle, préparez la politique gérée qui définit les autorisations qu'exige le rôle. Vous attachez cette politique au rôle dans une étape ultérieure.

   a. Dans le volet de navigation de gauche, choisissez Policies, puis Create Policy.

   b. En regard de Create Your Own Policy (Créez votre politique), choisissez Select (Sélectionner).

   c. Saisissez **`access-accountA-bucket`** dans le champ Policy Name.

   d. Copiez la stratégie d'accès suivante et collez-la dans le champ Policy Document. La politique d'accès accorde l'`s3:GetObject`autorisation de rôle. Ainsi, lorsque l'utilisateur du compte C assume le rôle, il ne peut effectuer que l'`s3:GetObject`opération.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::example-s3-bucket1/*"
       }
     ]
   }
   ```

   e. Choisissez Create Policy (Créer une politique).

   La nouvelle politique apparaît dans la liste des politiques gérées.

2. Dans le volet de navigation de gauche, choisissez Rôles, puis choisissez Créer un nouveau rôle.

3. Sous Sélectionner le type de rôle, sélectionnez Rôle pour l'accès entre comptes, puis cliquez sur le bouton Sélectionner à côté de Fournir un accès entre les Comptes AWS vôtres.

4. Saisissez l'ID de compte du Compte C.

   Pour cette procédure pas à pas, il n'est pas nécessaire de demander aux utilisateurs de disposer d'une authentification multifactorielle (MFA) pour assumer ce rôle. Ne sélectionnez donc pas cette option.

5. Choisissez Next Step pour définir les autorisations qui seront associées au rôle.
6. Cochez la case à côté de la politique Access-Accounta-Bucket que vous avez créée, puis choisissez Next Step.

   La page de révision apparaît pour que vous confirmiez les paramètres pour le rôle avant qu'il ne soit créé. Sur cette page, il est très important que vous notiez le lien que vous pouvez envoyer aux utilisateurs qui ont besoin d'utiliser ce rôle. Les utilisateurs qui utilisent le lien accèdent directement à la page Changer de rôle avec les champs ID de compte et Nom de rôle déjà remplis. Vous pouvez également consulter ce lien ultérieurement sur la page Récapitulatif des rôles pour tout rôle multicompte.
7. Entrez `examplerole` le nom du rôle, puis choisissez Next Step.
8. Après avoir examiné le rôle, choisissez Create Role.

   Le rôle `examplerole` est affiché dans la liste des rôles.
9. Choisissez le nom du rôle`examplerole`.
10. Sélectionnez l'onglet Trust Relationships.
11. Choisissez Afficher le document de stratégie et vérifiez que la politique de confiance affichée correspond à la politique suivante.

    La stratégie d'approbation suivante instaure la confiance avec le Compte C, en lui permettant d'exécuter l'action `sts:AssumeRole`. Pour plus d'informations, consultez AssumeRole dans la Référence d'API AWS Security Token Service .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountC-ID:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

12. Notez le nom de ressource Amazon (ARN) du `examplerole` rôle que vous avez créé.

Dans les étapes suivantes, vous attachez une stratégie d'utilisateur pour permettre un utilisateur IAM d'assumer ce rôle et vous identifiez le rôle par la valeur ARN.

Étape 2 : Tâches du compte B

L'exemple de compartiment détenu par le compte A a besoin d'objets détenus par d'autres comptes. Dans cette étape, l'administrateur du Compte B charge un objet grâce aux outils de ligne de commande.

- À l'aide de la `put-object` AWS CLI commande, chargez un objet vers*example-s3-bucket1*.

  ```
  aws s3api put-object --bucket example-s3-bucket1 --key HappyFace.jpg --
  body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucketOwner" --
  profile AccountBadmin
  ```

  Notez ce qui suit :

  - Le `--Profile` paramètre spécifie le `AccountBadmin` profil, de sorte que l'objet appartient au compte B.
  - Le paramètre `grant-full-control` accorde au propriétaire du compartiment l'autorisation de contrôle total sur l'objet comme l'exige la stratégie de compartiment.
  - Le paramètre `--body` identifie le fichier source à charger. Par exemple, si le fichier se trouve sur le lecteur C : d'un Windows ordinateur, vous devez le spécifier`c:\HappyFace.jpg`.

Étape 3 : Exécuter les tâches du compte C

Au cours des étapes précédentes, le compte A a déjà créé un rôle`examplerole`, établissant la confiance avec le compte C. Ce rôle permet aux utilisateurs du compte C d'accéder au compte A. À cette étape, l'administrateur du compte C crée un utilisateur (Dave) et lui délègue l'`sts:AssumeRole`autorisation qu'il a reçue du compte A. Cette approche permet à Dave d'assumer le compte `examplerole` et d'y accéder temporairement. La politique d'accès que le compte A a attachée au rôle limite ce que Dave peut faire lorsqu'il accède au compte A, en particulier, accéder à des objets. *example-s3-bucket1*

Étape 3.1 : créer un utilisateur dans le compte C et déléguer l'autorisation d'assumer examplerole

1. À l'aide de l'URL de connexion utilisateur IAM pour le compte C, connectez-vous d'abord au compte en AWS Management Console tant qu'**`AccountCadmin`**utilisateur.

2. Dans la console IAM, créez un utilisateur, Dave.

   Pour step-by-step obtenir des instructions, consultez la section Création d'utilisateurs IAM (AWS Management Console) dans le guide de l'utilisateur IAM.

3. Notez les informations d'identification de Dave. Dave a besoin de ces informations d'identification pour assumer le rôle `examplerole`.

4. Créez une politique intégrée permettant à l'utilisateur Dave IAM de déléguer l'`sts:AssumeRole`autorisation à Dave sur le `examplerole` rôle dans le compte A.

   a. Dans le panneau de navigation de gauche, sélectionnez Users (Utilisateurs).

   b. Choisissez le nom d'utilisateur Dave.

   c. Dans la page des détails de l'utilisateur, sélectionnez l'onglet Permissions (Autorisations) et développez la section Inline Policies (Stratégies en ligne).

   d. Choisissez click here (ou Create User Policy).

   e. Choisissez Custom Policy, puis Select.

   f. Saisissez un nom pour la stratégie dans le champ Policy Name.

   g. Copiez la stratégie suivante dans le champ Policy Document.

      Vous devez mettre à jour la politique en fournissant le*`AccountA-ID`*.

      ```
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": ["sts:AssumeRole"],
            "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
          }
        ]
      }
      ```

   h. Choisissez Apply Policy (Appliquer la stratégie).

5. Enregistrez les informations d'identification de Dave dans le fichier de configuration du AWS CLI en ajoutant un autre profil,`AccountCDave`.

```
[profile AccountCDave]
aws_access_key_id = UserDaveAccessKeyID
aws_secret_access_key = UserDaveSecretAccessKey
region = us-west-2
```

Étape 3.2 : Assumer le rôle (examplerole) et accéder aux objets

A présent, Dave peut accéder aux objets du compartiment détenu par le Compte A comme suit :

- Dave assume tout d'abord le rôle `examplerole` grâce à ses propres informations d'identification. Cela renvoie des autorisations temporaires.
- Grâce aux autorisations, Dave accède ensuite aux objets du compartiment du Compte A.

1. À l'invite de commande, exécutez la AWS CLI `assume-role` commande suivante à l'aide du `AccountCDave` profil.

   Vous devez mettre à jour la valeur ARN dans la commande en indiquant *`AccountA-ID`* où `examplerole` est défini.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile
 AccountCDave --role-session-name test
```

   En réponse, AWS Security Token Service (AWS STS) renvoie des informations de sécurité temporaires (ID de clé d'accès, clé d'accès secrète et jeton de session).

2. Enregistrez les informations de sécurité temporaires dans le fichier de AWS CLI configuration situé sous le `TempCred` profil.

```
[profile TempCred]
aws_access_key_id = temp-access-key-ID
aws_secret_access_key = temp-secret-access-key
aws_session_token = session-token
region = us-west-2
```

3. À l'invite de commande, exécutez la AWS CLI commande suivante pour accéder aux objets à l'aide des informations d'identification temporaires. Par exemple, la commande spécifie l'API d'objet HEAD pour récupérer les métadonnées d'objet pour l'objet `HappyFace.jpg`.

```
aws s3api get-object --bucket example-s3-bucket1 --key HappyFace.jpg SaveFileAs.jpg
 --profile TempCred
```

Etant donné que la stratégie d'accès attachée au rôle `examplerole` autorise les actions, Amazon S3 traite la demande. Vous pouvez essayer n'importe quelle autre action sur n'importe quel objet du compartiment.

Si vous essayez une autre action, par exemple, l'autorisation `get-object-acl` vous sera refusée car le rôle n'est pas autorisé à effectuer cette action.

```
aws s3api get-object-acl --bucket example-s3-bucket1 --key HappyFace.jpg --profile
 TempCred
```

Nous avons utilisé Dave pour assumer le rôle et accéder à l'objet grâce à des autorisations temporaires. Il peut s'agir également d'une application dans le Compte C qui accède aux objets dans le compartiment *example-s3-bucket1*. L'application peut obtenir des informations d'identification de sécurité temporaires, et le Compte C peut déléguer l'autorisation d'application pour assumer le rôle `examplerole`.

Étape 4 : Nettoyer

1. Une fois les tests terminés, vous pouvez effectuer les opérations suivantes pour effectuer le nettoyage :

   - Connectez-vous à la AWS Management Console en utilisant les autorisations du compte A, et procédez comme suit :

     - Dans la console Amazon S3, supprimez la stratégie de compartiment attachée à examplebucket. Dans la page Properties du compartiment, supprimez la stratégie dans la section Permissions.
     - Si le compartiment a été créé pour cet exercice, supprimez les objets, puis le compartiment, dans la console Amazon S3.

- Dans la console IAM, supprimez le `examplerole` que vous avez créé dans le compte A. Pour step-by-step obtenir des instructions, reportez-vous à la section Suppression d'un utilisateur IAM dans le guide de l'utilisateur IAM.
- Dans la console IAM, supprimez l'AccountAadminutilisateur.

2. Connectez-vous à la console IAM à l'aide des informations d'identification du compte B. Supprimez l'utilisateur AccountBadmin.
3. Connectez-vous à la console IAM à l'aide des informations d'identification du compte C. Supprimer AccountCadminet l'utilisateur Dave.

Ressources connexes

Pour plus d'informations relatives à cette procédure pas à pas, consultez les ressources suivantes dans le guide de l'utilisateur IAM :

- Création d'un rôle pour déléguer des autorisations à un utilisateur IAM
- Tutoriel : Déléguer l'accès à Comptes AWS l'aide de rôles IAM
- Gestion des politiques IAM

Comment Amazon S3 autorise une demande

Lorsqu'Amazon S3 reçoit une demande : par exemple, une opération de compartiment ou d'objet, il vérifie que le demandeur possède les autorisations nécessaires. Amazon S3 évalue toutes les politiques d'accès, les politiques utilisateur et les politiques basées sur les ressources pertinentes (politique des compartiments, liste de contrôle d'accès aux compartiments (ACL) et ACL des objets) pour décider d'autoriser ou non la demande.

> **Note**
>
> Si le contrôle des autorisations Amazon S3 ne trouve pas d'autorisations valides, une erreur d'autorisation d'accès refusé (403 Interdit) est renvoyée. Pour plus d'informations, consultez Résoudre les erreurs d'accès refusé (403 Interdit) dans Amazon S3.

Pour déterminer si le demandeur est autorisé à effectuer l'opération spécifique, Amazon S3 effectue les opérations suivantes, dans l'ordre, lorsqu'il reçoit une demande :

1. Convertit toutes les politiques d'accès pertinentes (politique utilisateur, politique de compartiment et ACL) au moment de l'exécution en un ensemble de politiques à évaluer.
2. Evalue l'ensemble de stratégies obtenu au cours des étapes suivantes. Dans chaque étape, Amazon S3 évalue un sous-ensemble de stratégies dans un contexte spécifique, basé sur l'autorité du contexte.
   a. Contexte d'utilisateur – Dans le contexte d'utilisateur, le compte parent auquel appartient l'utilisateur est l'autorité du contexte.

      Amazon S3 évalue un sous-ensemble de stratégies détenues par le compte parent. Ce sous-ensemble inclut la stratégie d'utilisateur que le parent attache à l'utilisateur. Si le parent possède également la ressource contenue dans la demande (compartiment ou objet), Amazon S3 évalue également les politiques de ressources correspondantes (politique de compartiment, ACL de compartiment et ACL d'objet) en même temps.

      Un utilisateur doit avoir l'autorisation du compte parent pour exécuter l'opération.

      Cette étape s'applique uniquement si la demande est faite par un utilisateur dans un Compte AWS. Si la demande est faite à l'aide des informations d'identification de l'utilisateur root d'un Compte AWS, Amazon S3 ignore cette étape.

b. Contexte du compartiment : dans le contexte du compartiment, Amazon S3 évalue les politiques détenues par Compte AWS le propriétaire du compartiment.

   Si la demande concerne une opération de compartiment, le demandeur doit avoir l'autorisation du propriétaire du compartiment. Si la demande concerne un objet, Amazon S3 évalue toutes les stratégies détenues par le propriétaire du compartiment pour vérifier que ce dernier n'a pas refusé explicitement l'accès à l'objet. Si un refus explicite est configuré, Amazon S3 n'autorise pas la demande.

c. Contexte d'objet – Si la demande concerne un objet, Amazon S3 évalue le sous-ensemble de stratégies détenues par le propriétaire de l'objet.

Voici quelques exemples de scénarios qui illustrent la manière dont Amazon S3 autorise une demande.

Example — Le demandeur est un IAM principal

Si le demandeur est un principal IAM, Amazon S3 doit déterminer si le parent auquel Compte AWS appartient le principal a accordé au principal l'autorisation nécessaire pour effectuer l'opération. De plus, si la demande concerne une opération de compartiment, comme une demande pour lister le contenu du compartiment, Amazon S3 doit vérifier que le propriétaire du compartiment a accordé l'autorisation au demandeur d'exécuter l'opération. Pour effectuer une opération spécifique sur une ressource, un principal IAM doit obtenir l'autorisation du parent Compte AWS auquel il appartient et du Compte AWS propriétaire de la ressource.

Example — Le demandeur est un IAM principal — Si la demande concerne une opération sur un objet dont le propriétaire du bucket n'est pas propriétaire

Si la demande concerne une opération sur un objet dont le propriétaire du compartiment n'est pas propriétaire, Amazon S3 doit non seulement s'assurer que le demandeur dispose des autorisations du propriétaire de l'objet, mais également vérifier la politique du compartiment afin de s'assurer que le propriétaire du compartiment n'a pas défini de refus explicite pour l'objet. Le propriétaire du compartiment (qui paie la facture) peut refuser explicitement l'accès aux objets dans le compartiment, quel que soit le propriétaire. Le propriétaire du compartiment peut également supprimer tout objet du compartiment

Par défaut, lorsqu'un autre Compte AWS utilisateur télécharge un objet dans votre compartiment S3, ce compte (le rédacteur de l'objet) est propriétaire de l'objet, y a accès et peut autoriser d'autres utilisateurs à y accéder via des listes de contrôle d'accès (ACL). Vous pouvez utiliser

Object Ownership afin de modifier ce comportement par défaut, pour que les ACL soient désactivées et que vous, en tant que propriétaire du compartiment, possédiez automatiquement tous les objets de votre compartiment. Par conséquent, le contrôle d'accès à vos données est basé sur des politiques, telles que les politiques utilisateur IAM, les politiques relatives aux compartiments S3, les politiques relatives aux points de terminaison du cloud privé virtuel (VPC) AWS Organizations et les politiques de contrôle des services (SCP). Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Pour en savoir plus sur la façon dont Amazon S3 évalue les stratégies d'accès pour autoriser ou refuser les demandes d'opérations de compartiment et d'objets, consultez les rubriques suivantes :

Rubriques

- Comment Amazon S3 autorise une demande pour une opération de compartiment
- Comment Amazon S3 autorise une demande pour une opération sur les objets

Comment Amazon S3 autorise une demande pour une opération de compartiment

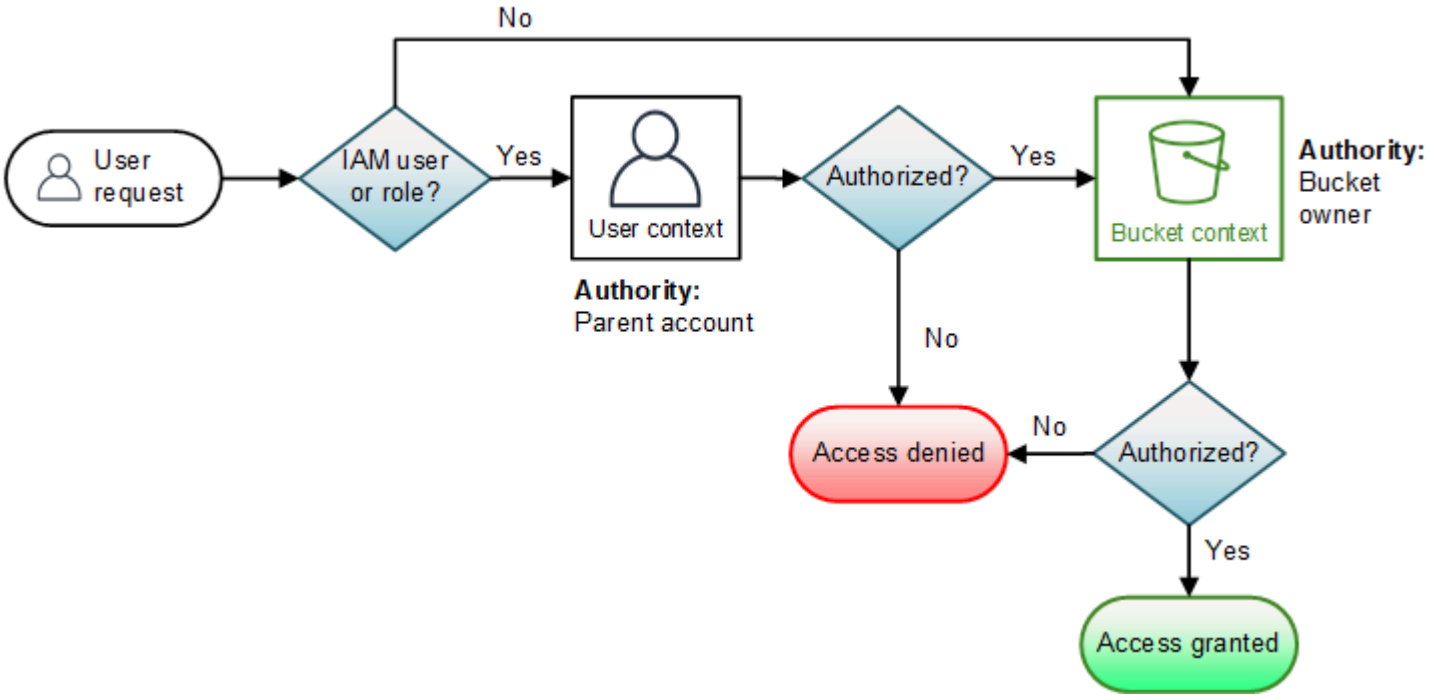
Lorsqu'Amazon S3 reçoit une demande d'opération de compartiment, Amazon S3 convertit toutes les autorisations pertinentes en un ensemble de stratégies à évaluer lors de l'exécution. Les autorisations pertinentes incluent les autorisations basées sur les ressources (par exemple, les stratégies de compartiment et les listes de contrôle d'accès des compartiments) et les stratégies utilisateur si la demande provient d'un principal IAM. Amazon S3 évalue ensuite l'ensemble de politiques qui en résulte en une série d'étapes en fonction d'un contexte spécifique (contexte utilisateur ou contexte de compartiment) :

1. Contexte utilisateur — Si le demandeur est un principal IAM, le principal doit avoir l'autorisation du parent Compte AWS auquel il appartient. Dans cette étape, Amazon S3 évalue un sous-ensemble de stratégies détenues par le compte parent (également appelé autorité du contexte). Ce sous-ensemble de stratégies inclut la stratégie d'utilisateur que le compte parent attache au principal. Si le parent détient également la ressource de la demande (dans ce cas, le compartiment), Amazon S3 évalue également les stratégies de ressources correspondantes (stratégie de compartiment et liste ACL de compartiment) en même temps. Lorsqu'une demande pour une opération de compartiment est faite, les journaux d'accès au serveur enregistre l'ID canonique du demandeur. Pour plus d'informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.
2. Contexte de compartiment – Le demandeur doit avoir l'autorisation du propriétaire du compartiment pour exécuter une opération de compartiment spécifique. Au cours de cette étape,

Amazon S3 évalue un sous-ensemble de politiques appartenant au propriétaire du Compte AWS compartiment.

Le propriétaire du compartiment peut accorder l'autorisation grâce à la stratégie de compartiment ou la liste ACL de compartiment. Si le Compte AWS propriétaire du bucket est également le compte parent d'un IAM principal, il peut configurer les autorisations du bucket dans une politique utilisateur.

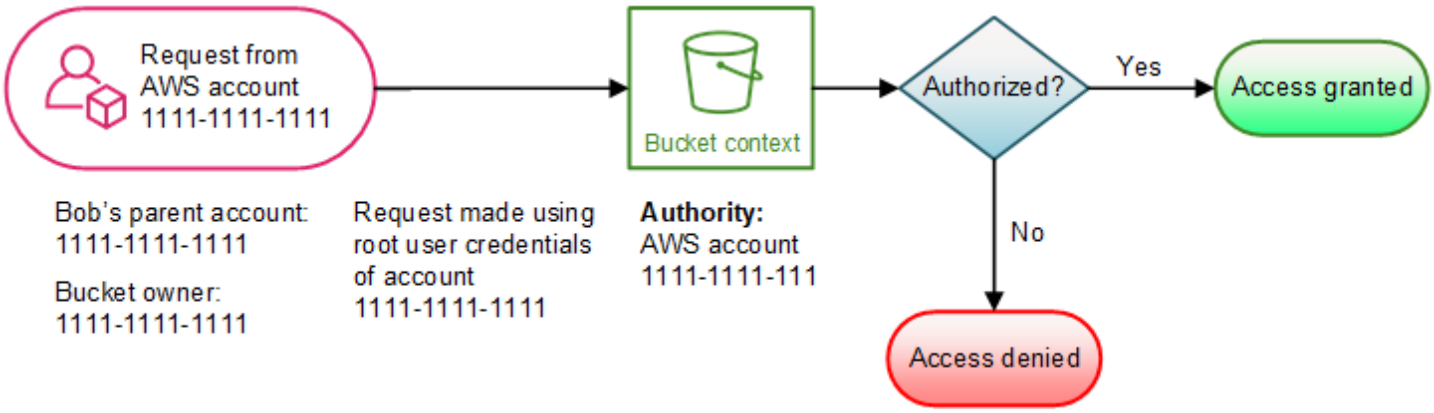
Voici un schéma de l'évaluation basée sur le contexte pour les opérations de compartiment.



Les exemples suivants illustrent la logique d'évaluation.

Exemple 1 : Opération de compartiment demandée par le propriétaire du compartiment

Dans cet exemple, le propriétaire du compartiment envoie une demande pour une opération de compartiment grâce aux informations d'identification racine du Compte AWS.

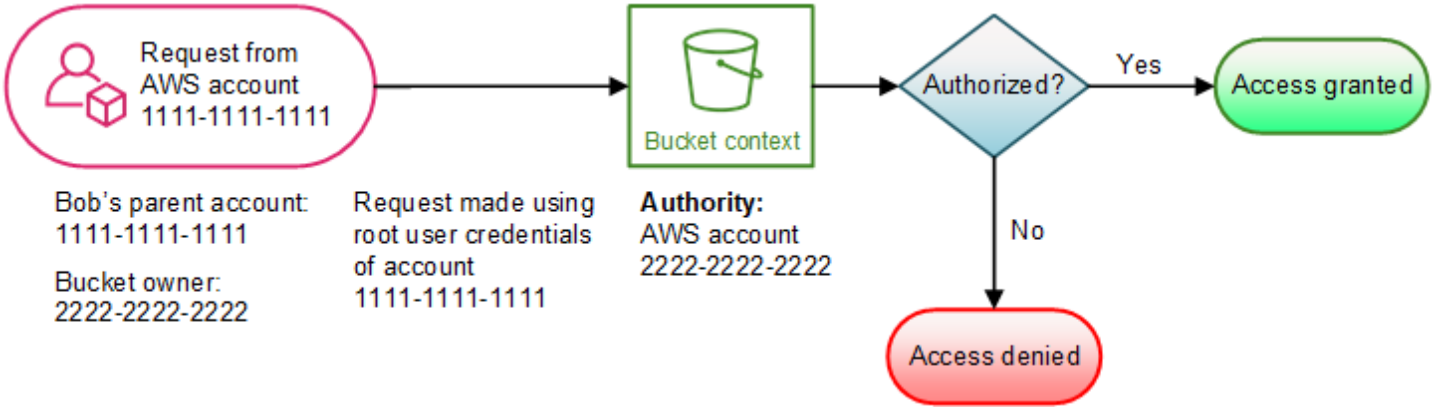


Amazon S3 réalise l'évaluation du contexte comme suit :

1. Étant donné que la demande a été faite grâce aux informations d'identification d'utilisateur root d'un Compte AWS, le contexte d'utilisateur n'est pas évalué.
2. Dans le contexte de compartiment, Amazon S3 examine la stratégie de compartiment pour déterminer si le demandeur a l'autorisation d'exécuter l'opération. Amazon S3 autorise la demande.

Exemple 2 : opération de compartiment demandée par une personne Compte AWS qui n'est pas le propriétaire du compartiment

Dans cet exemple, une demande est faite grâce aux informations d'identification d'utilisateur root du Compte AWS 1111-1111-1111 pour une opération de compartiment détenue par le Compte AWS 2222-2222-2222. Aucun utilisateur IAM n'est impliqué dans cette demande.



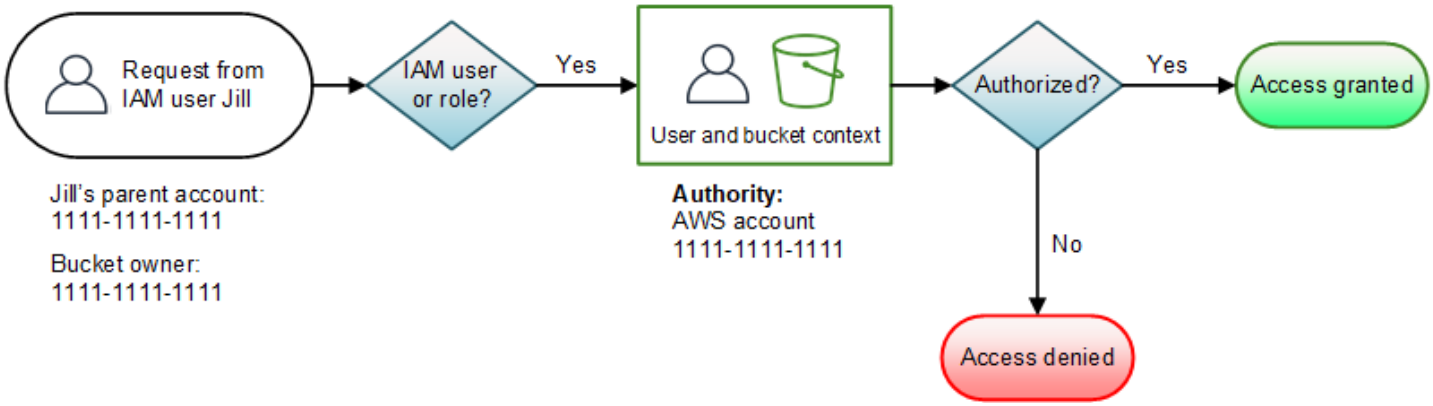
Dans cet exemple, Amazon S3 évalue le contexte comme suit :

1. Comme la demande est faite à l'aide des informations d'identification de l'utilisateur root d'un Compte AWS, le contexte utilisateur n'est pas évalué.

2. Dans le contexte de compartiment, Amazon S3 examine la stratégie de compartiment. Si le propriétaire du compartiment (Compte AWS 2222-2222-2222) n'a pas autorisé le Compte AWS 1111-1111-1111 à effectuer l'opération demandée, Amazon S3 refuse la demande. Sinon, Amazon S3 accorde la demande et exécute l'opération.

Exemple 3 : opération de compartiment demandée par un directeur IAM dont le parent Compte AWS est également le propriétaire du compartiment

Dans l'exemple, la demande est envoyée par Jill, une utilisatrice IAM du Compte AWS 1111-1111-1111, qui détient également le compartiment.



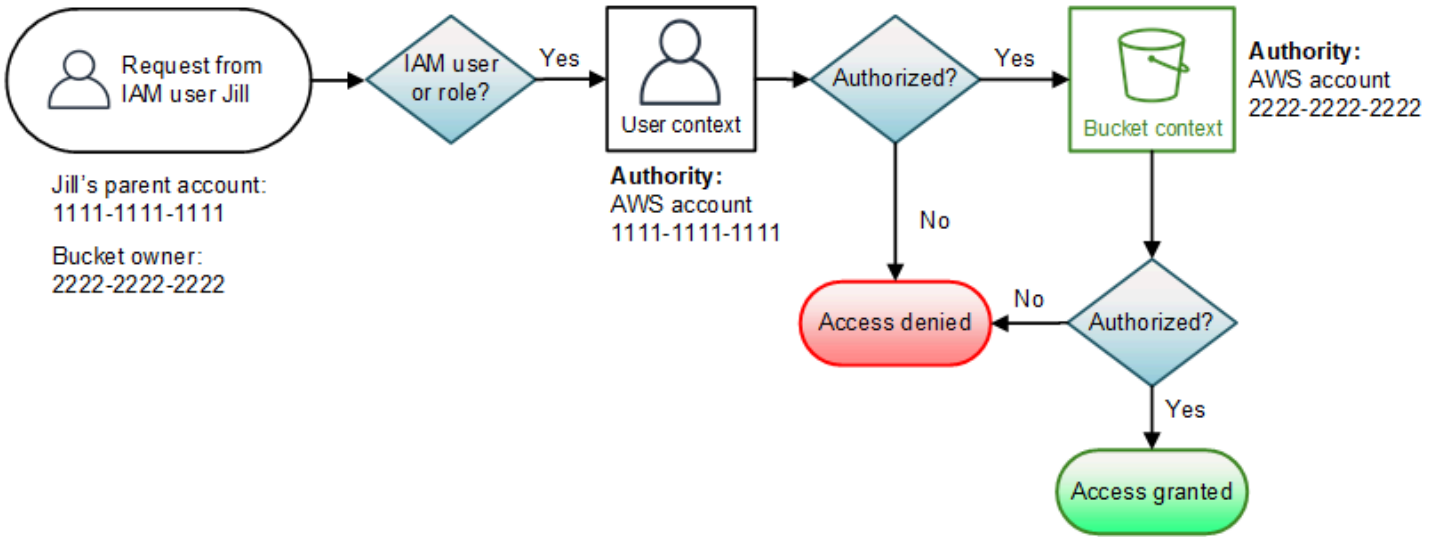
Amazon S3 réalise l'évaluation suivante du contexte :

1. Comme la demande provient d'un principal IAM, dans le contexte de l'utilisateur, Amazon S3 évalue toutes les politiques appartenant au parent Compte AWS afin de déterminer si Jill est autorisée à effectuer l'opération.

   Dans cet exemple, le parent Compte AWS 1111-1111-1111, auquel appartient le principal, est également le propriétaire du compartiment. Par conséquent, en plus de la politique utilisateur, Amazon S3 évalue également la politique du compartiment et l'ACL du compartiment dans le même contexte, car elles appartiennent au même compte.
2. Amazon S3 a évalué la stratégie de compartiment et la liste ACL de compartiment dans le contexte d'utilisateur, il n'a dont pas évalué le contexte de compartiment.

Exemple 4 : opération de compartiment demandée par un principal IAM dont le parent n' Compte AWS est pas le propriétaire du compartiment

Dans cet exemple, la demande est envoyée par Jill, une utilisatrice IAM dont le parent Compte AWS est 1111-1111-1111, mais le bucket appartient à un autre utilisateur, 2222-2222-2222. Compte AWS



Jill aura besoin des autorisations du parent Compte AWS et du propriétaire du compartiment. Amazon S3 évalue le contexte comme suit :

1. Étant donné que la demande provient d'un principal IAM, Amazon S3 évalue le contexte d'utilisateur en examinant les stratégies autorisées par le compte pour vérifier que Jill possède les autorisations nécessaires. Si Jill a l'autorisation, Amazon S3 passe à l'évaluation du contexte du compartiment. Si Jill n'a pas l'autorisation, elle refuse la demande.
2. Dans le contexte du compartiment, Amazon S3 vérifie que le propriétaire du compartiment 2222-2222-2222 a accordé à Jill (ou à ses parents Compte AWS) l'autorisation d'effectuer l'opération demandée. Si elle dispose de cette autorisation, Amazon S3 accepte la demande et exécute l'opération. Sinon, Amazon S3 refuse la demande.

Comment Amazon S3 autorise une demande pour une opération sur les objets

Lorsqu'Amazon S3 reçoit une demande pour une opération d'objet, il convertit toutes les autorisations pertinentes, comme les autorisations basées sur les ressources (liste de contrôle d'accès d'objet, stratégie de compartiment, liste de contrôle d'accès (ACL) de compartiment) et les stratégies d'utilisateur IAM, en un ensemble de stratégies à évaluer au moment de l'exécution. Ensuite, il évalue l'ensemble de stratégies obtenu au cours d'une série d'étapes. À chaque étape, il évalue un sous-ensemble de politiques dans trois contextes spécifiques : le contexte utilisateur, le contexte du compartiment et le contexte de l'objet :

1. Contexte utilisateur — Si le demandeur est un principal IAM, le principal doit avoir l'autorisation du parent Compte AWS auquel il appartient. Dans cette étape, Amazon S3 évalue un sous-ensemble de stratégies détenues par le compte parent (également appelé autorité du contexte). Ce sous-

ensemble de stratégies inclut la stratégie d'utilisateur que le parent attache au principal. Si le parent possède également la ressource contenue dans la demande (compartiment ou objet), Amazon S3 évalue les politiques de ressources correspondantes (politique de compartiment, ACL de compartiment et ACL d'objet) en même temps.

> **Note**
>
> Si le parent Compte AWS est propriétaire de la ressource (compartiment ou objet), il peut accorder des autorisations de ressource à son principal IAM en utilisant la politique utilisateur ou la politique de ressource.

2. Contexte de compartiment – Dans ce contexte, Amazon S3 évalue les stratégies détenues par le Compte AWS propriétaire du compartiment.

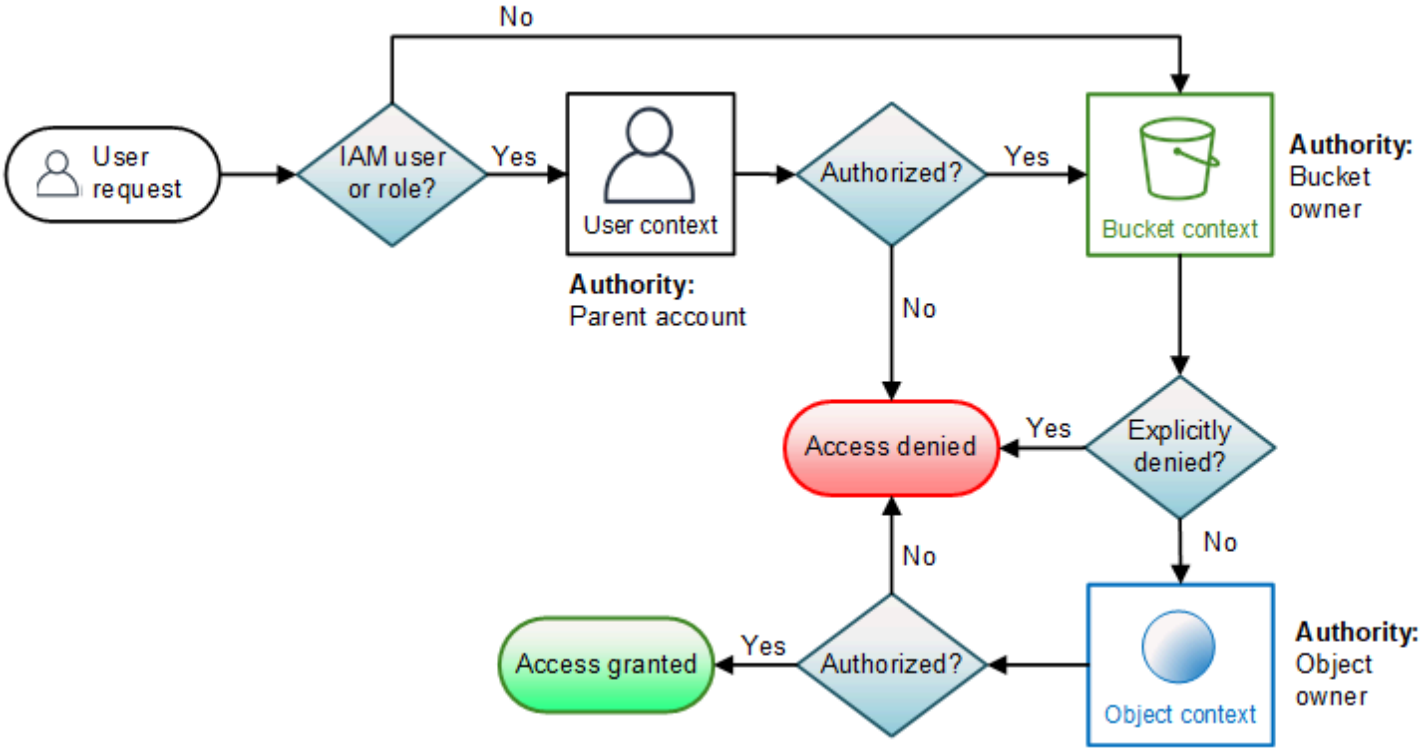
   Si le Compte AWS propriétaire de l'objet indiqué dans la demande n'est pas le même que le propriétaire du compartiment, Amazon S3 vérifie les politiques si le propriétaire du compartiment a explicitement refusé l'accès à l'objet. Si un refus explicite est configuré sur l'objet, Amazon S3 n'autorise pas la demande.

3. Contexte d'objet – Le demandeur doit avoir l'autorisation du propriétaire de l'objet pour exécuter une opération d'objet spécifique. Dans cette étape, Amazon S3 évalue la liste ACL d'objet.

> **Note**
>
> Si les propriétaires du compartiment et de l'objet sont les mêmes, l'accès à l'objet peut être autorisé dans la stratégie de compartiment, qui est évaluée dans le contexte de compartiment. Si les propriétaires sont différents, les propriétaires de l'objet doivent utiliser une liste ACL d'objet pour accorder les autorisations. Si le Compte AWS propriétaire de l'objet est également le compte parent auquel appartient le principal IAM, celui-ci peut configurer les autorisations de l'objet dans une politique utilisateur, qui est évaluée dans le contexte de l'utilisateur. Pour en savoir plus sur l'utilisation de ces alternatives de stratégie d'accès, consultez Procédures pas à pas utilisant des politiques pour gérer l'accès à vos ressources Amazon S3.
>
> Si, en tant que propriétaire du bucket, vous souhaitez être propriétaire de tous les objets de votre bucket et utiliser des politiques de bucket ou des politiques basées sur IAM pour gérer l'accès à ces objets, vous pouvez appliquer le paramètre imposé par le propriétaire du bucket pour la propriété des objets. Avec ce paramètre, en tant que propriétaire du compartiment, vous possédez automatiquement tous les objets de votre compartiment et

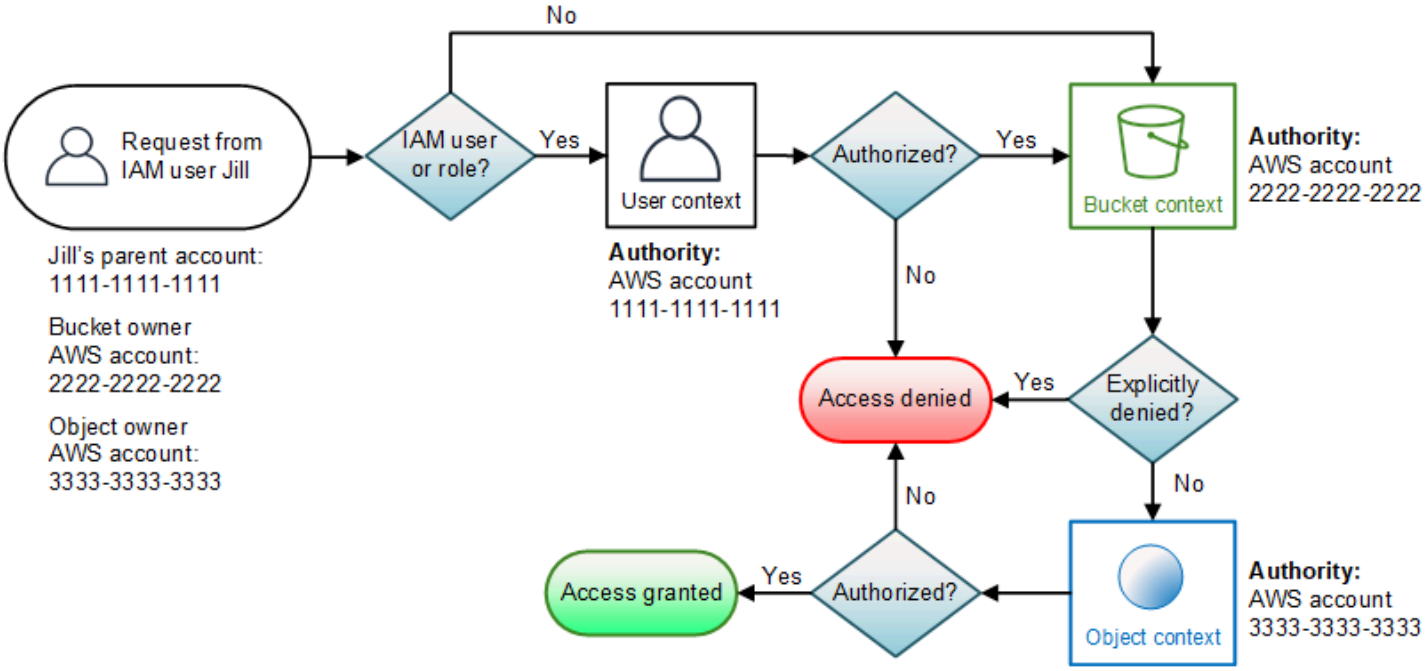
avez le contrôle total sur chaque objet présents dans votre compartiment. Les listes ACL de compartiment et d'objet ne peuvent pas être modifiées et ne sont plus prises en compte pour y accéder. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Voici un schéma de l'évaluation basée sur le contexte pour les opérations d'objet.



Exemple de demande d'opération sur un objet

Dans cet exemple, l'utilisateur IAM Jill, dont le parent Compte AWS est 1111-1111-1111, envoie une demande d'opération d'objet (par exemple,`GetObject`) pour un objet appartenant à Compte AWS 3333-3333-3333 dans un bucket appartenant au 2222-2222-2222. Compte AWS



Jill aura besoin de l'autorisation du parent Compte AWS, du propriétaire du bucket et du propriétaire de l'objet. Amazon S3 évalue le contexte comme suit :

1. Comme la demande provient d'un principal IAM, Amazon S3 évalue le contexte utilisateur pour vérifier que le parent Compte AWS 1111-1111-1111 a autorisé Jill à effectuer l'opération demandée. Si Jill a l'autorisation, Amazon S3 évalue le contexte de compartiment. Sinon, Amazon S3 refuse la demande.
2. Dans le contexte du compartiment, le propriétaire du compartiment, Compte AWS 2222-2222-2222, est l'autorité du contexte. Amazon S3 évalue la stratégie de compartiment pour déterminer si le propriétaire du compartiment a explicitement refusé que Jill accède à l'objet.
3. Dans le contexte de l'objet, l'autorité du contexte est le Compte AWS 3333-3333-3333, le propriétaire de l'objet. Amazon S3 évalue la liste ACL d'objet pour déterminer si Jill a l'autorisation d'accéder à l'objet. Si oui, Amazon S3 autorise la demande.

AWS politiques gérées pour Amazon S3

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez la section Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS politique gérée : AmazonS3FullAccess

Vous pouvez attacher la politique `AmazonS3FullAccess` à vos identités IAM. Cette politique accorde des autorisations qui permettent un accès complet à Amazon S3.

Pour consulter les autorisations relatives à cette politique, consultez AmazonS3FullAccess dans AWS Management Console.

AWS politique gérée : AmazonS3ReadOnlyAccess

Vous pouvez attacher la politique `AmazonS3ReadOnlyAccess` à vos identités IAM. Cette politique accorde des autorisations qui permettent d'accéder en lecture seule à Amazon S3.

Pour consulter les autorisations relatives à cette politique, consultez AmazonS3ReadOnlyAccess dans AWS Management Console.

Politique gérée par AWS : AmazonS3ObjectLambdaExecutionRolePolicy

Fournit aux AWS Lambda fonctions les autorisations requises pour envoyer des données à S3 Object Lambda lorsque des demandes sont adressées à un point d'accès S3 Object Lambda. Accorde également à Lambda l'autorisation d'écrire dans les journaux Amazon CloudWatch .

Pour consulter les autorisations relatives à cette politique, consultez AmazonS3ObjectLambdaExecutionRolePolicy dans AWS Management Console.

Amazon S3 met à jour les politiques AWS gérées

Consultez les informations relatives aux mises à jour des politiques AWS gérées pour Amazon S3 depuis que ce service a commencé à suivre ces modifications.

Modification	Description	Date
Amazon S3 a ajouté des autorisations Describe à `AmazonS3ReadOnlyAccess`	Amazon S3 a ajouté des autorisations `s3:Describe*` à `AmazonS3ReadOnlyAccess` .	11 août 2023
Amazon S3 a ajouté des autorisations S3 Object Lambda à `AmazonS3FullAccess` et `AmazonS3ReadOnlyAccess` .	Amazon S3 a mis à jour les politiques `AmazonS3FullAccess` et `AmazonS3ReadOnlyAccess` pour inclure les autorisations pour S3 Object Lambda.	27 septembre 2021
Amazon S3 a ajouté `AmazonS3ObjectLambdaExecutionRolePolicy`	Amazon S3 a ajouté une nouvelle politique AWS gérée appelée `AmazonS3ObjectLambdaExecutionRolePolicy` qui fournit aux fonctions Lambda des autorisations leur permettant d'interagir avec S3 Object Lambda et d'écrire dans des journaux. CloudWatch	18 août 2021
Amazon S3 a commencé à assurer le suivi des modifications	Amazon S3 a commencé à suivre les modifications apportées AWS à ses politiques gérées.	18 août 2021

Utilisation des rôles liés à un service pour le cadre de stockage Amazon S3

Pour utiliser Amazon S3 Storage Lens afin de collecter et agréger des mesures sur tous vos comptes dans les AWS Organizations, vous devez d'abord vous assurer que S3 Storage Lens dispose d'un accès fiable activé par le compte de gestion de votre organisation. S3 Storage Lens crée un rôle lié à un service (SLR) pour lui permettre d'obtenir la liste des Comptes AWS membres de votre organisation. Cette liste de comptes est utilisée par S3 Storage Lens pour collecter des mesures des ressources S3 dans tous les comptes membres lorsque le tableau de bord ou les configurations de S3 Storage Lens sont créés ou mis à jour.

Amazon S3 Storage Lens utilise des AWS Identity and Access Management rôles liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement au S3 Storage Lens. Les rôles liés au service sont prédéfinis par S3 Storage Lens et incluent toutes les autorisations dont le service a besoin pour appeler d'autres personnes en votre Services AWS nom.

Un rôle lié à un service simplifie la configuration de S3 Storage Lens car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. S3 Storage Lens définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul S3 Storage Lens peut endosser ses rôles. Les autorisations définies comprennent la politique d’approbation et la politique d’autorisation. De plus, cette politique d’autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer le rôle lié à un service uniquement après avoir supprimé les ressources connexes. Vos ressources relatives au S3 Storage Lens sont ainsi protégées, car vous ne pouvez pas accidentellement supprimer les autorisations d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez AWS services that work with IAM (Services AWS fonctionnant avec IAM) et recherchez les services avec un Yes (Oui) dans la colonne Service-Linked Role (Rôle lié à un service). Choisissez un Oui ayant un lien permettant de consulter la documentation du rôle lié à un service, pour ce service.

Autorisations de rôle liées au service pour le cadre de stockage Amazon S3

S3 Storage Lens utilise le rôle lié au service nommé AWSServiceRoleForS3StorageLens— Cela permet d'accéder aux AWS services et aux ressources utilisés ou gérés par S3 Storage Lens. Cela permet à S3 Storage Lens d'accéder aux AWS Organizations ressources en votre nom.

Le rôle lié à un service S3 Storage Lens approuve le service suivant sur le stockage de votre organisation :

- `storage-lens.s3.amazonaws.com`

La stratégie d'autorisations liée au rôle permet au S3 Storage Lens de réaliser les actions suivantes :

- `organizations:DescribeOrganization`

  `organizations:ListAccounts`

  `organizations:ListAWSServiceAccessForOrganization`

  `organizations:ListDelegatedAdministrators`

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour S3 Storage Lens

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous effectuez l'une des tâches suivantes alors que vous êtes connecté aux comptes AWS Organizations de gestion ou d'administrateur délégué, S3 Storage Lens crée le rôle lié au service pour vous :

- Créez une configuration de tableau de bord S3 Storage Lens pour votre organisation dans la console Amazon S3.
- PUTune configuration S3 Storage Lens pour votre organisation à l'aide de l'API REST AWS CLI et des SDK.

> **Note**
>
> S3 Storage Lens prendra en charge un maximum de cinq administrateurs délégués par organisation.

Si vous supprimez ce rôle lié à un service, les actions précédentes le recréeront si nécessaire.

Exemple de stratégie pour un rôle lié à un service de S3 Storage Lens

Example Stratégie d'autorisations pour le rôle lié à un service de S3 Storage Lens

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AwsOrgsAccess",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Modification d'un rôle lié à un service pour Amazon S3 Storage Lens

S3 Storage Lens ne vous permet pas de modifier le rôle AWSServiceRoleForS3StorageLens lié au service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le IAM Guide de l'utilisateur.

Suppression d'un rôle lié à un service pour Amazon S3 Storage Lens

Si vous n'utilisez plus le rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le service Amazon S3 Storage Lens utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer le, AWSServiceRoleForS3StorageLens vous devez supprimer toutes les configurations S3 Storage Lens présentes au niveau de l'organisation à Régions AWS l'aide des comptes de AWS Organizations gestion ou d'administrateur délégué.

Les ressources sont des configurations de S3 Storage Lens au niveau de l'organisation. Utilisez S3 Storage Lens pour nettoyer les ressources, puis utilisez la console IAM, la CLI, l'API REST ou le AWS SDK pour supprimer le rôle.

Dans l'API REST et les SDK AWS CLI, les configurations S3 Storage Lens peuvent être découvertes `ListStorageLensConfigurations` dans toutes les régions dans lesquelles votre organisation a créé des configurations S3 Storage Lens. Utilisez l'action `DeleteStorageLensConfiguration` pour supprimer ces configurations afin de pouvoir ensuite supprimer le rôle.

> **Note**
>
> Pour supprimer le rôle lié au service, vous devez supprimer toutes les configurations de S3 Storage Lens au niveau de l'organisation dans toutes les Régions où elles existent.

Pour supprimer les ressources Amazon S3 Storage Lens utilisées par le AWSServiceRoleForS3StorageLens SLR

1. Pour obtenir une liste des configurations au niveau de votre organisation, vous devez utiliser le `ListStorageLensConfigurations` dans chaque région dans laquelle vous disposez de configurations S3 Storage Lens. Cette liste peut également être obtenue à partir de la console Amazon S3.
2. Supprimez ces configurations des points de terminaison régionaux appropriés en appelant l'appel d'`DeleteStorageLensConfiguration`API ou en utilisant la console Amazon S3.

Pour supprimer manuellement le rôle lié à un service à l’aide d’IAM

Après avoir supprimé les configurations, supprimez le AWSServiceRoleForS3StorageLens SLR de la console IAM, en invoquant l'API IAM ou en utilisant `DeleteServiceLinkedRole` le SDK. AWS CLI AWS Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions supportées pour les rôles liés à un service de S3 Storage Lens

S3 Storage Lens prend en charge l'utilisation de rôles liés au service partout Régions AWS où le service est disponible. Pour plus d'informations, veuillez consulter Régions et points de terminaison Amazon S3.

Résolution des problèmes d'identité et d'accès à Amazon S3

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec Amazon S3 et IAM.

Rubriques

- J'ai reçu un message d'erreur de refus d'accès
- Je ne suis pas autorisé à effectuer une action dans Amazon S3
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures Compte AWS à moi à accéder à mes ressources Amazon S3

J'ai reçu un message d'erreur de refus d'accès

Vérifiez qu'il n'existe aucune `Deny` déclaration explicite contre le demandeur auquel vous essayez d'accorder des autorisations, que ce soit dans la politique de compartiment ou dans la politique basée sur l'identité.

Pour obtenir des informations détaillées sur la résolution des erreurs de refus d'accès, consultezRésolution des erreurs d'accès refusé (403 interdit) dans Amazon S3.

Je ne suis pas autorisé à effectuer une action dans Amazon S3

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `s3:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 s3:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `s3:GetWidget`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à Amazon S3.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans Amazon S3. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures Compte AWS à moi à accéder à mes ressources Amazon S3

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques

basées sur les ressources ou les listes de contrôle d’accès (ACL), vous pouvez utiliser ces politiques pour donner l’accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si Amazon S3 prend en charge ces fonctionnalités, consultezComment Amazon S3 fonctionne avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d’identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d’identité) dans le Guide de l’utilisateur IAM.
- Pour connaître la différence entre l'utilisation de rôles et de politiques basées sur les ressources pour l'accès entre comptes, consultez la section Accès aux ressources entre comptes dans IAM dans le guide de l'utilisateur d'IAM.

Gestion de l’accès avec les octrois d’accès S3

Pour respecter le principe du moindre privilège, vous définissez un accès granulaire à vos données Amazon S3 en fonction des applications, des personas, des groupes ou des unités organisationnelles. Vous pouvez utiliser diverses approches pour mettre en place un accès granulaire aux données dans Amazon S3, en fonction de l’échelle et de la complexité des modèles d’accès.

L'approche la plus simple pour gérer l'accès à un small-to-medium certain nombre d'ensembles de données dans Amazon S3 par les principaux AWS Identity and Access Management (IAM) consiste à définir des politiques d'autorisation IAM et des politiques de compartiment S3. Cette stratégie fonctionne, à condition que les politiques nécessaires respectent les limites de taille des politiques de compartiment S3 (20 Ko) et des politiques IAM (5 Ko), ainsi que le nombre de principaux IAM autorisés par compte.

Avec l’augmentation du nombre de jeux de données et de cas d’utilisation, il se peut que vous ayez besoin de plus d’espace de politique. Une approche qui offre beaucoup plus d’espace pour les déclarations de politique consiste à utiliser les points d’accès S3 comme points de terminaison

supplémentaires pour les compartiments S3, car chaque point d'accès peut avoir sa propre politique. Vous pouvez définir des modèles de contrôle d'accès très précis, car vous pouvez avoir des milliers de points d'accès Région AWS par compte, avec une politique d'une taille maximale de 20 Ko pour chaque point d'accès. Bien que les points d'accès S3 augmentent la quantité d'espace de politique disponible, ils nécessitent un mécanisme permettant aux clients de découvrir le point d'accès approprié pour le jeu de données approprié.

Une troisième approche consiste à implémenter un modèle d'agent de session IAM, dans lequel vous implémentez une logique de décision d'accès et générez dynamiquement des informations d'identification de session IAM à court terme pour chaque session d'accès. L'approche de l'agent de session IAM prend en charge les modèles d'autorisations arbitrairement dynamiques et s'adapte efficacement, mais vous devez élaborer la logique des modèles d'accès.

Au lieu d'utiliser ces approches, vous pouvez utiliser les octrois d'accès S3 pour gérer l'accès à vos données Amazon S3. Les octrois d'accès S3 fournissent un modèle simplifié pour définir les autorisations d'accès aux données dans Amazon S3 par préfixe, compartiment ou objet. En outre, vous pouvez utiliser les octrois d'accès S3 pour accorder l'accès à la fois aux principaux IAM et directement aux utilisateurs ou groupes depuis votre annuaire d'entreprise.

Vous définissez généralement les autorisations d'accès aux données dans Amazon S3 en mappant les utilisateurs et les groupes aux jeux de données. Vous pouvez utiliser les octrois d'accès S3 pour définir des mappages d'accès direct des préfixes S3 aux utilisateurs et aux rôles au sein des compartiments et des objets Amazon S3. Grâce au schéma d'accès simplifié des octrois d'accès S3, vous pouvez accorder un accès en lecture seule, en écriture seule ou en lecture-écriture en fonction d'un préfixe S3 à la fois aux principaux IAM et directement aux utilisateurs ou groupes depuis un annuaire d'entreprise. Grâce à ces fonctionnalités d'octrois d'accès S3, les applications peuvent demander des données à Amazon S3 au nom de l'utilisateur authentifié actuel de l'application.

Lorsque vous intégrez les subventions d'accès S3 à la fonctionnalité de propagation d'identité sécurisée de AWS IAM Identity Center, vos applications peuvent envoyer des demandes Services AWS (y compris les subventions d'accès S3) directement au nom d'un utilisateur authentifié de l'annuaire d'entreprise. Vos applications n'ont plus besoin de commencer par mapper l'utilisateur à un principal IAM. En outre, étant donné que les identités des utilisateurs finaux sont propagées jusqu'à Amazon S3, l'audit permettant de vérifier quel utilisateur a accédé à quel objet S3 est simplifié. Il n'est plus nécessaire de reconstruire la relation entre les différents utilisateurs et les sessions IAM. Lorsque vous utilisez les octrois d'accès S3 avec la propagation d'identité approuvée d'IAM Identity Center, chaque événement de données AWS CloudTrail pour Amazon S3 contient une référence directe à l'utilisateur final pour le compte duquel les données ont été consultées.

Pour plus d’informations sur les octrois d’accès S3, consultez les rubriques suivantes.

Rubriques

Concepts des octrois d’accès S3

Les octrois d’accès S3 introduisent les concepts suivants pour leur schéma d’accès simplifié :

Instances d’octrois d’accès S3

Une instance d’octrois d’accès S3 est un conteneur logique pour les octrois individuels qui définissent qui dispose de quel niveau d’accès à quelles données Amazon S3. Vous pouvez avoir une instance d’octrois d’accès S3 par Région AWS par Compte AWS. Vous utilisez cette instance S3 Access Grants pour contrôler l'accès à tous les compartiments d'un même compte et Région AWS. Si vous souhaitez utiliser S3 Access Grants pour accorder l'accès aux identités d'utilisateurs et de groupes dans votre annuaire d'entreprise, vous devez également associer votre instance S3 Access Grants à une instance AWS Identity and Access Management (IAM) Identity Center.

Emplacements

Un emplacement définit les données auxquelles votre instance d’octrois d’accès S3 peut accorder l’accès. Les octrois d’accès S3 fonctionnent en distribuant des informations d’identification IAM

dont l'accès est limité à un préfixe, un compartiment ou un objet S3 particulier. Vous associez un emplacement d'octrois d'accès S3 à un rôle IAM, à partir duquel ces sessions temporaires sont créées. La configuration d'emplacement la plus courante est un emplacement unique dans `s3://` pour l'ensemble de l'instance d'octrois d'accès S3, qui peut couvrir l'accès à tous les compartiments S3 du compte et de la Région AWS. Vous pouvez également créer plusieurs emplacements dans votre instance d'octrois d'accès S3. Par exemple, vous pouvez enregistrer un compartiment comme emplacement `s3://example-s3-bucket1` pour les octrois que vous souhaitez limiter à ce compartiment, et vous pouvez également enregistrer l'emplacement par défaut `s3://`.

Octrois

Pour réduire la portée de l'accès au sein d'un emplacement, vous créez des octrois individuels. Un octroi individuel dans une instance d'octrois d'accès S3 permet à une entité spécifique (un principal IAM ou un utilisateur ou un groupe dans un annuaire d'entreprise) d'accéder à un préfixe, un compartiment ou un objet Amazon S3. Pour chaque octroi, vous pouvez définir une portée (un préfixe, un compartiment ou un objet) et un niveau d'accès (`READ`, `WRITE` ou `READWRITE`) différents. Par exemple, vous pouvez avoir un octroi qui autorise un groupe d'annuaires d'entreprise particulier, `01234567-89ab-cdef-0123-456789abcdef` `READ` à accéder à `s3://example-s3-bucket1/projects/items/*`. Cet octroi donne aux utilisateurs de ce groupe l'accès `READ` à tous les objets dont le nom de clé présente le préfixe `projects/items/` dans le compartiment nommé *`example-s3-bucket1`*.

Informations d'identification temporaires des octrois d'accès S3

Une application peut demander des informations d' just-in-time accès en appelant une nouvelle opération d'API S3 GetDataAccess, pour demander l'accès à un seul objet, préfixe ou compartiment avec un niveau d'autorisation de `READWRITE`, ou`READWRITE`. L'instance d'octrois d'accès S3 évalue la demande `GetDataAccess` par rapport aux octrois dont elle dispose. S'il existe un octroi correspondant, les octrois d'accès S3 endossent le rôle IAM associé à l'emplacement de l'octroi correspondant. Les octrois d'accès S3 délimitent ensuite les autorisations de la session IAM précisément au compartiment, préfixe ou objet S3 spécifié par l'étendue de l'octroi. Le délai d'expiration des informations d'accès temporaires est par défaut de 1 heure, mais vous pouvez le définir sur une valeur comprise entre 15 minutes et 12 heures.

Comment ça marche

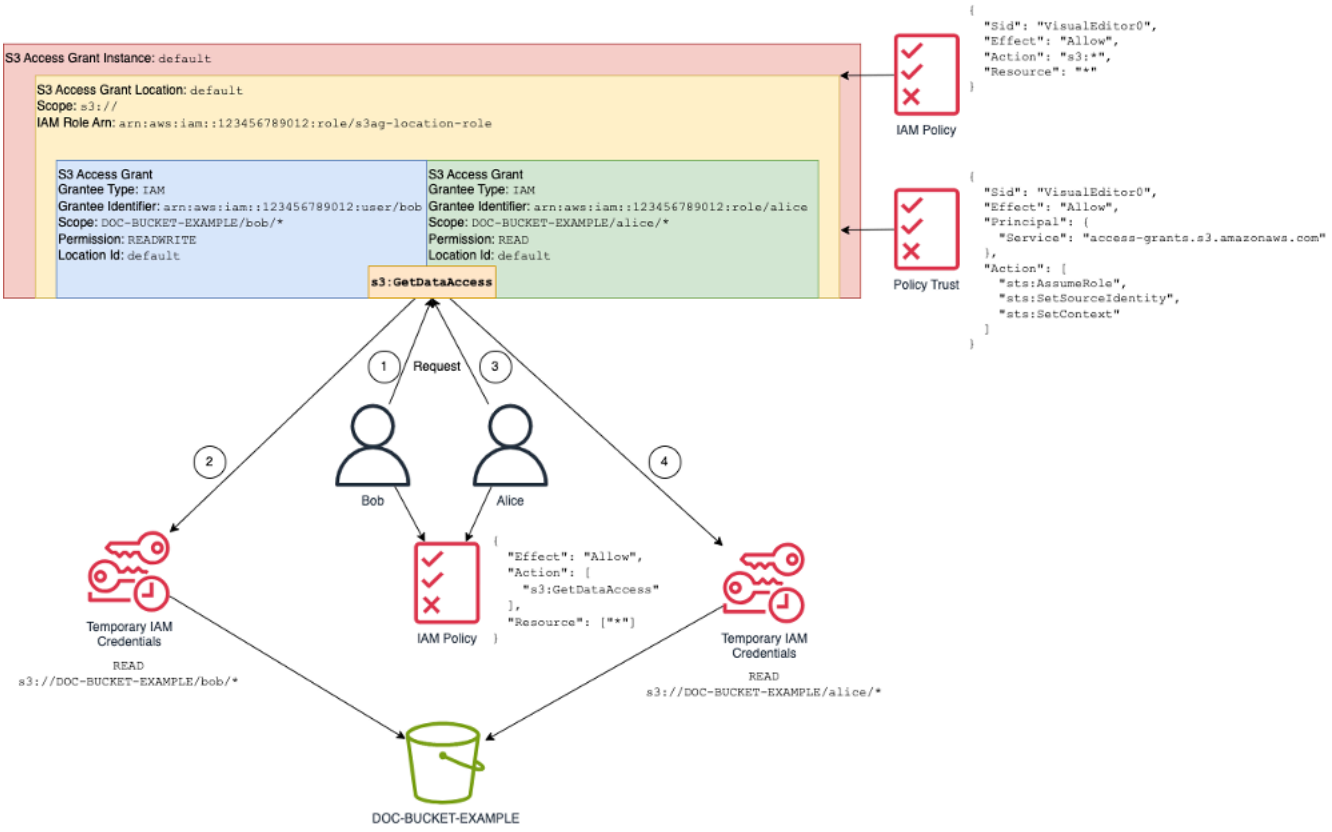
Dans le schéma suivant, un emplacement Amazon S3 par défaut avec la ^portée `s3://` est enregistré avec le rôle IAM `s3ag-location-role`. Ce rôle IAM est autorisé à effectuer des actions

Amazon S3 dans le compte lorsque ses informations d'identification sont obtenues via les octrois d'accès S3.

Au sein de cet emplacement, deux octrois d'accès individuels sont créés pour deux utilisateurs IAM. L'utilisateur IAM Bob se voit octroyer l'accès `READ` et l'accès `WRITE` sur le préfixe `bob/` dans le compartiment `DOC-BUCKET-EXAMPLE`. Un autre rôle IAM, Alice, n'est autorisé `READ` à accéder qu'au `alice/` préfixe du `DOC-BUCKET-EXAMPLE` compartiment. Un octroi, coloré en bleu, est défini pour permettre à Bob d'accéder au préfixe `bob/` dans le compartiment `DOC-BUCKET-EXAMPLE`. Un octroi, coloré en vert, est défini pour permettre à Alice d'accéder au préfixe `alice/` dans le compartiment `DOC-BUCKET-EXAMPLE`.

Lorsqu'il est temps pour Bob d'accéder aux `READ` données, le rôle IAM associé à l'emplacement dans lequel se trouve sa subvention appelle l'opération d'GetDataAccessAPI S3 Access Grants. Si Bob essaie de lire (`READ`) un préfixe ou un objet S3 quelconque commençant par `s3://DOC-BUCKET-EXAMPLE/bob/*`, la demande `GetDataAccess` renvoie un ensemble d'informations d'identification de session IAM temporaires avec l'autorisation à `s3://DOC-BUCKET-EXAMPLE/bob/*`. De même, Bob peut écrire (`WRITE`) sur n'importe quel préfixe ou objet S3 commençant par `s3://DOC-BUCKET-EXAMPLE/bob/*`, car l'octroi le permet également.

De même, Alice peut lire (`READ`) tout ce qui commence par `s3://DOC-BUCKET-EXAMPLE/alice/`. Toutefois, si elle essaie d'écrire (`WRITE`) sur un compartiment, un préfixe ou un objet quelconque dans `s3://`, elle recevra une erreur Accès refusé (403 – Interdit), car aucun octroi ne lui donne l'accès `WRITE` sur aucune donnée. En outre, si Alice demande un niveau d'accès quelconque (`READ` ou `WRITE`) à des données situées en dehors de `s3://DOC-BUCKET-EXAMPLE/alice/`, elle recevra à nouveau une erreur Accès refusé.



Ce modèle s’adapte à un nombre élevé d’utilisateurs et de compartiments, et simplifie la gestion de ces autorisations. Au lieu de modifier des politiques de compartiment S3 potentiellement volumineuses chaque fois que vous souhaitez ajouter ou supprimer une relation individuelle d’accès par préfixe utilisateur, vous pouvez ajouter et supprimer des octrois discrets et individuels.

Octrois d’accès S3 et identités d’annuaire d’entreprise

Vous pouvez utiliser les subventions d'accès Amazon S3 pour accorder l'accès aux principaux AWS Identity and Access Management (utilisateurs ou rôles) (IAM), à la fois dans le même environnement Compte AWS et dans d'autres. Toutefois, dans de nombreux cas, l’entité accédant aux données est un utilisateur final issu de votre annuaire d’entreprise. Au lieu d’accorder l’accès aux principaux IAM, vous pouvez utiliser les octrois d’accès S3 pour accorder l’accès directement aux utilisateurs et aux groupes de votre entreprise. Avec les octrois d’accès S3, vous n’avez plus besoin de mapper vos identités d’entreprise à des principaux IAM intermédiaires pour accéder à vos données S3 via vos applications d’entreprise.

Cette nouvelle fonctionnalité (prise en charge de l'utilisation des identités des utilisateurs finaux pour accéder aux données) est fournie en associant votre instance S3 Access Grants à une instance. AWS IAM Identity Center IAM Identity Center prend en charge les fournisseurs d'identité basés

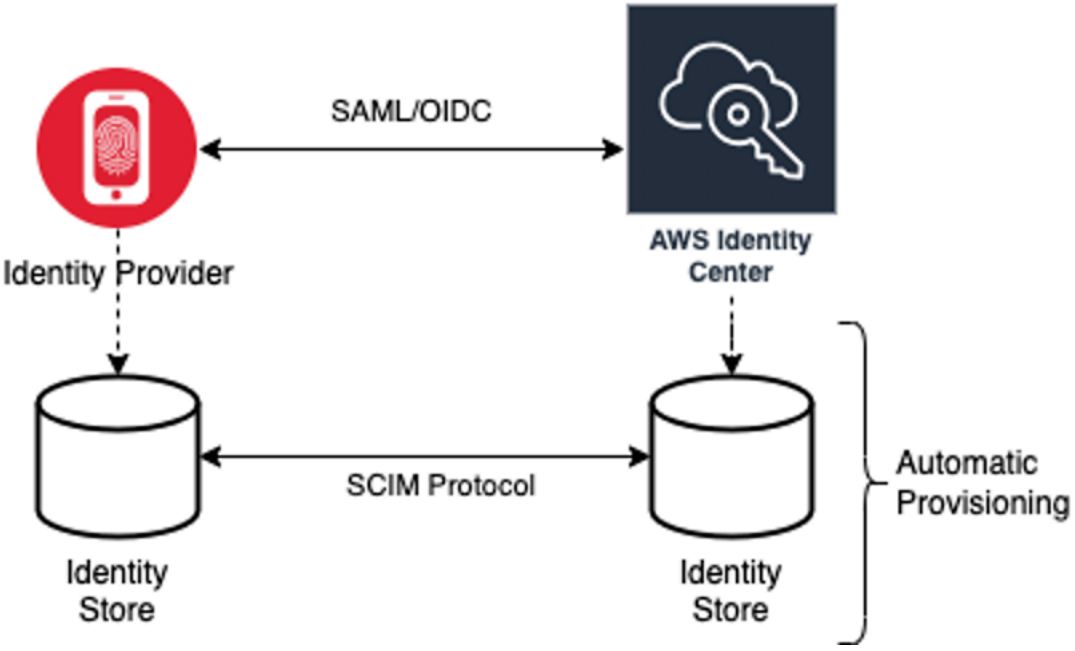
sur des normes et constitue la plaque tournante AWS pour tous les services ou fonctionnalités, y compris les subventions d'accès S3, qui prennent en charge les identités des utilisateurs finaux. IAM Identity Center fournit un support d'authentification pour les identités d'entreprise via sa fonctionnalité de propagation d'identité approuvée. Pour plus d'informations, consultez Propagation d'identité approuvée entre applications.

Pour commencer à prendre en charge l'identité du personnel dans les octrois d'accès S3, vous devez configurer au préalable le provisionnement des identités entre votre fournisseur d'identité d'entreprise et IAM Identity Center dans IAM Identity Center. IAM Identity Center prend en charge les fournisseurs d'identité d'entreprise tels que Okta, Microsoft Entra ID (anciennement Azure Active Directory) ou tout autre fournisseur d'identité (IdP) externe prenant en charge le protocole de mise en service du système de gestion des identités inter-domaines (SCIM). Lorsque vous connectez IAM Identity Center à votre fournisseur d'identité et que vous activez le provisionnement automatique, les utilisateurs et les groupes de votre IdP sont synchronisés dans le magasin d'identités d'IAM Identity Center. Après cette étape, IAM Identity Center dispose de sa propre vision de vos utilisateurs et de vos groupes, de sorte que vous pouvez y faire référence en utilisant d'autres Services AWS fonctionnalités, telles que S3 Access Grants. Pour plus d'informations sur la configuration du provisionnement automatique d'IAM Identity Center, consultez Provisionnement automatique dans le Guide de l'utilisateur AWS IAM Identity Center .

IAM Identity Center est intégré AWS Organizations afin que vous puissiez gérer de manière centralisée les autorisations sur plusieurs comptes Comptes AWS sans configurer manuellement chacun de vos comptes. Dans une organisation standard, votre administrateur d'identité configure une seule instance IAM Identity Center pour l'organisation tout entière, en tant que point unique de synchronisation des identités. Cette instance IAM Identity Center s'exécute généralement dans une instance dédiée Compte AWS de votre organisation. Dans cette configuration courante, vous pouvez faire référence aux identités des utilisateurs et des groupes dans S3 Access Grants depuis n'importe quel Compte AWS membre de l'organisation.

Toutefois, si votre AWS Organizations administrateur n'a pas encore configuré d'instance centrale d'IAM Identity Center, vous pouvez en créer une locale sur le même compte que votre instance S3 Access Grants. Une telle configuration est plus courante pour les cas proof-of-concept d'utilisation liés au développement local. Dans tous les cas, l'instance IAM Identity Center doit être Région AWS identique à l'instance S3 Access Grants à laquelle elle sera associée.

Dans le schéma suivant d'une configuration IAM Identity Center avec un IdP externe, le fournisseur d'identité est configuré avec SCIM pour synchroniser le magasin d'identités de l'IdP avec le magasin d'identités dans IAM Identity Center.



Pour utiliser vos identités d'annuaire d'entreprise avec les octrois d'accès S3, procédez comme suit :

- Configurez le provisionnement automatique dans IAM Identity Center pour synchroniser les informations relatives aux utilisateurs et aux groupes entre votre fournisseur d'identité et IAM Identity Center.
- Configurez votre source d'identité externe dans IAM Identity Center en tant qu'émetteur de jetons approuvé. Pour plus d'informations, consultez Propagation d'identité approuvée entre applications dans le Guide de l'utilisateur AWS IAM Identity Center .
- Associez votre instance d'octrois d'accès S3 à votre instance IAM Identity Center. Vous pouvez le faire lorsque vous créez votre instance d'octrois d'accès S3. Si vous avez déjà créé votre instance d'octrois d'accès S3, consultez Association ou dissociation de votre instance IAM Identity Center.

Accès des identités d'annuaire aux données S3

Supposons que vous ayez des utilisateurs d'annuaire d'entreprise qui ont besoin d'accéder à vos données S3 via une application d'entreprise, par exemple, une visionneuse de documents intégrée à votre fournisseur d'identité externe (par exemple, Okta) pour authentifier les utilisateurs. L'authentification de l'utilisateur dans ces applications se fait généralement par le biais de redirections dans le navigateur Web de l'utilisateur. Comme les utilisateurs figurant dans l'annuaire ne sont pas des principaux IAM, votre application a besoin d'informations d'identification IAM pour appeler l'opération d'API `GetDataAccess` des octrois d'accès S3 afin d'obtenir des informations d'identification d'accès aux données S3 au nom des utilisateurs. Contrairement aux utilisateurs et aux

rôles IAM qui obtiennent eux-mêmes des informations d'identification, votre application a besoin d'un moyen de représenter un utilisateur d'annuaire non mappé à un rôle IAM, afin que l'utilisateur puisse accéder aux données via les octrois d'accès S3.

Cette transition, d'un utilisateur d'annuaire authentifié à un appelant IAM capable d'adresser des demandes aux octrois d'accès S3 au nom de l'utilisateur d'annuaire, est effectuée par l'application via la fonctionnalité d'émetteur de jetons approuvé d'IAM Identity Center. Après avoir authentifié l'utilisateur d'annuaire, l'application dispose d'un jeton d'identité provenant du fournisseur d'identité (par exemple, Okta) qui représente l'utilisateur d'annuaire selon Okta. La configuration de l'émetteur de jetons approuvé dans IAM Identity Center permet à l'application d'échanger ce jeton Okta (le locataire Okta est configuré comme « émetteur approuvé ») contre un autre jeton d'identité issu d'IAM Identity Center qui représentera de manière sécurisée l'utilisateur d'annuaire au sein des Services AWS. L'application de données endosse alors un rôle IAM, fournissant le jeton de l'utilisateur d'annuaire provenant d'IAM Identity Center comme contexte supplémentaire. L'application peut utiliser la session IAM qui en résulte pour appeler des octrois d'accès S3. Le jeton représente à la fois l'identité de l'application (le principal IAM lui-même) et l'identité de l'utilisateur d'annuaire.

L'étape principale de cette transition est l'échange de jetons. L'application effectue cet échange de jetons en appelant l'opération d'API `CreateTokenWithIAM` dans IAM Identity Center. Bien entendu, il s'agit également d'un appel d' AWS API qui nécessite la signature d'un directeur IAM. Le principal IAM qui effectue cette demande est généralement un rôle IAM associé à l'application. Par exemple, si l'application s'exécute sur Amazon EC2, la demande `CreateTokenWithIAM` est généralement effectuée par le rôle IAM associé à l'instance EC2 sur laquelle l'application s'exécute. Le résultat d'un `CreateTokenWithIAM` appel réussi est un nouveau jeton d'identité, qui sera reconnu à l'intérieur Services AWS.

L'étape suivante, avant que l'application puisse appeler `GetDataAccess` au nom de l'utilisateur d'annuaire, consiste pour l'application à obtenir une session IAM incluant l'identité de l'utilisateur d'annuaire. Pour ce faire, l'application utilise une `AssumeRole` demande AWS Security Token Service (AWS STS) qui inclut également le jeton IAM Identity Center pour l'utilisateur de l'annuaire en tant que contexte d'identité supplémentaire. Ce contexte supplémentaire permet à IAM Identity Center de propager l'identité de l'utilisateur d'annuaire à l'étape suivante. Le rôle IAM endossé par l'application est celui qui nécessitera des autorisations IAM pour appeler l'opération `GetDataAccess`.

Après avoir endossé le rôle IAM du porteur d'identité avec le jeton IAM Identity Center pour l'utilisateur d'annuaire comme contexte supplémentaire, l'application dispose désormais de tout ce

dont elle a besoin pour adresser une demande signée à `GetDataAccess` au nom de l'utilisateur d'annuaire authentifié.

La propagation des jetons se base sur les étapes suivantes :

Création d'une application IAM Identity Center

Commencez par créer une nouvelle application dans IAM Identity Center. Cette application utilisera un modèle permettant à IAM Identity Center d'identifier le type de paramètres d'application que vous pouvez utiliser. La commande pour créer l'application nécessite que vous fournissiez l'Amazon Resource Name (ARN) de l'instance IAM Identity Center, un nom d'application et l'ARN du fournisseur d'application. Le fournisseur d'application est le fournisseur d'application SAML ou OAuth que l'application utilisera pour effectuer des appels à IAM Identity Center.

Pour utiliser l'exemple de commande suivant, remplacez les *`user input placeholders`* par vos propres informations :

```
aws sso-admin create-application \
 --instance-arn "arn:aws:sso:::instance/ssoins-ssoins-1234567890abcdef" \
 --application-provider-arn "arn:aws:sso::aws:applicationProvider/custom" \
 --name MyDataApplication
```

Réponse :

```
{
   "ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
```

Création d'un émetteur de jetons approuvé

Maintenant que vous disposez d'une application IAM Identity Center, l'étape suivante consiste à configurer un émetteur de jetons approuvé qui permettra d'échanger vos valeurs `IdToken` de votre fournisseur d'identité contre des jetons IAM Identity Center. Dans cette étape, vous devez fournir les éléments suivants :

- URL de l'émetteur du fournisseur d'identité
- Nom de l'émetteur de jetons approuvé
- Chemin d'attribut de demande
- Chemin d'attribut de magasin d'identités

- Option de récupération de l'ensemble de clés web JSON (JWKS)

Le chemin d'attribut de demande est l'attribut du fournisseur d'identité qui sera utilisé pour la mappage à l'attribut de magasin d'identités. Normalement, le chemin d'attribut de demande est l'adresse e-mail de l'utilisateur, mais vous pouvez utiliser d'autres attributs pour effectuer le mappage.

Créez un fichier appelé `oidc-configuration.json` avec les informations suivantes. Pour utiliser ce fichier, remplacez les *`user input placeholders`* par vos propres informations.

```
{
  "OidcJwtConfiguration":
     {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
     }
}
```

Pour créer l'émetteur de jetons approuvé, exécutez la commande suivante. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws sso-admin create-trusted-token-issuer \
  --instance-arn "arn:aws:sso:::instance/ssoins-1234567890abcdef" \
  --name MyEntraIDTrustedIssuer \
  --trusted-token-issuer-type OIDC_JWT \
  --trusted-token-issuer-configuration file://./oidc-configuration.json
```

Réponse

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
}
```

Connexion de l'application IAM Identity Center à l'émetteur de jetons approuvé

L'émetteur de jetons approuvé a besoin de quelques paramètres de configuration supplémentaires pour fonctionner. Définissez l'audience à laquelle l'émetteur de jetons approuvé fera confiance.

L'audience est la valeur figurant dans `IdToken` qui est identifiée par la clé et qui se trouve dans les paramètres du fournisseur d'identité. Par exemple :

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Créez un fichier nommé `grant.json` qui contient le contenu suivant. Pour utiliser ce fichier, modifiez l'audience afin qu'elle corresponde aux paramètres de votre fournisseur d'identité et fournissez l'ARN de l'émetteur de jetons approuvé renvoyé par la commande précédente.

```
{
   "JwtBearer":
     {
       "AuthorizedTokenIssuers":
         [
           {
             "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
               "AuthorizedAudiences":
                 [
                   "1234973b-abcd-1234-abcd-345c5a9c1234"
                 ]
           }
         ]
     }
 }
```

Exécutez l'exemple de commande suivant. Pour utiliser cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws sso-admin put-application-grant \
  --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
  --grant-type "urn:ietf:params:oauth:grant-type:jwt-bearer" \
  --grant file://./grant.json \
```

Cette commande définit les paramètres de configuration de l'émetteur de jetons approuvé pour qu'il fasse confiance à l'audience définie dans le fichier `grant.json` et relie cette audience à l'application créée à la première étape pour échanger des jetons du type `jwt-bearer`. La chaîne `urn:ietf:params:oauth:grant-type:jwt-bearer` n'est pas une chaîne arbitraire. Il s'agit

d'un espace de noms enregistré dans les profils d'assertion de jeton Web JSON (JWT) OAuth. Vous trouverez plus d'informations sur cet espace de noms dans la RFC 7523.

Ensuite, utilisez la commande suivante pour définir les portées que l'émetteur de jetons approuvé inclura lors de l'échange de valeurs `IdToken` à partir de votre fournisseur d'identité. Pour les octrois d'accès S3, la valeur du paramètre `--scope` est `s3:access_grants:read_write`.

```
aws sso-admin put-application-access-scope \
  --application-arn "arn:aws:sso::111122223333:application/ssoins-
ssoins-111122223333abcdef/apl-abcd1234a1b2c3d" \
  --scope "s3:access_grants:read_write"
```

La dernière étape consiste à attacher une politique de ressources à l'application IAM Identity Center. Cette politique permettra au rôle IAM de votre application d'adresser des demandes à l'opération d'API `sso-oauth:CreateTokenWithIAM` et de recevoir les valeurs `IdToken` d'IAM Identity Center.

Créez un fichier nommé `authentication-method.json` qui contient le contenu suivant. Remplacez *123456789012* par votre ID de compte.

```
{
   "Iam":
       {
         "ActorPolicy":
             {
                "Version": "2012-10-17",
                    "Statement":
                    [
                        {
                           "Effect": "Allow",
                            "Principal":
                            {
                              "AWS": "arn:aws:iam::123456789012:role/webapp"
                            },
                           "Action": "sso-oauth:CreateTokenWithIAM",
                            "Resource": "*"
                        }
                    ]
                }
            }
        }
```

Pour attacher la politique à l'application IAM Identity Center, exécutez la commande suivante :

```
aws sso-admin put-application-authentication-method \
   --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
   --authentication-method-type IAM \
   --authentication-method file://./authentication-method.json
```

Ceci complète les paramètres de configuration pour l'utilisation des octrois d'accès S3 avec les utilisateurs d'annuaire via une application Web. Vous pouvez tester cette configuration directement dans l'application ou vous pouvez appeler l'opération d'API `CreateTokenWithIAM` en utilisant la commande suivante à partir d'un rôle IAM autorisé dans la politique d'application IAM Identity Center :

```
aws sso-oidc create-token-with-iam \
   --client-id "arn:aws:sso::123456789012:application/ssoins-ssoins-1234567890abcdef/
apl-abcd1234a1b2c3d"  \
   --grant-type urn:ietf:params:oauth:grant-type:jwt-bearer \
   --assertion IdToken
```

La réponse sera similaire à ceci :

```
{
    "accessToken": "<suppressed long string to reduce space>",
    "tokenType": "Bearer",
    "expiresIn": 3600,
    "refreshToken": "<suppressed long string to reduce space>",
    "idToken": "<suppressed long string to reduce space>",
    "issuedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",
    "scope": [
      "sts:identity_context",
      "s3:access_grants:read_write",
      "openid",
      "aws"
    ]
}
```

Si vous décodez la valeur `IdToken` codée en base64, vous pouvez voir les paires clé-valeur au format JSON. La clé `sts:identity_context` contient la valeur que votre application doit envoyer dans la demande `sts:AssumeRole` pour inclure les informations d'identité de l'utilisateur d'annuaire. Voici un exemple de valeur `IdToken` décodée :

```
{
    "aws:identity_store_id": "d-996773e796",
    "sts:identity_context": "AQoJb3JpZ2luX2VjEOTtl;<SUPRESSED>",
    "sub": "83d43802-00b1-7054-db02-f1d683aacba5",
    "aws:instance_account": "123456789012",
    "iss": "https://identitycenter.amazonaws.com/ssoins-1234567890abcdef",
    "sts:audit_context": "AQoJb3JpZ2luX2VjEOT<SUPRESSED>==",
    "aws:identity_store_arn": "arn:aws:identitystore::232642235904:identitystore/
d-996773e796",
    "aud": "abcd12344U0gi7n4Yyp0-WV1LWNlbnRyYWwtMQ",
    "aws:instance_arn": "arn:aws:sso:::instance/ssoins-6987d7fb04cf7a51",
    "aws:credential_id": "EXAMPLEHI5glPh40y9TpApJn8...",
    "act": {
       "sub": "arn:aws:sso::232642235904:trustedTokenIssuer/
ssoins-6987d7fb04cf7a51/43b4a822-1020-7053-3631-cb2d3e28d10e"
    },
    "auth_time": "2023-11-01T20:24:28Z",
    "exp": 1698873868,
    "iat": 1698870268
}
```

Vous pouvez obtenir la valeur à partir de `sts:identity_context` et transmettre ces informations dans un appel `sts:AssumeRole`. Vous trouverez ci-dessous un exemple CLI de la syntaxe. Le rôle à endosser est un rôle temporaire avec des autorisations pour invoquer `s3:GetDataAccess`.

```
aws sts assume-role \
   --role-arn "arn:aws:iam::123456789012:role/temp-role" \
   --role-session-name "TempDirectoryUserRole" \
   --provided-contexts ProviderArn="arn:aws:iam::aws:contextProvider/
IdentityCenter",ContextAssertion="value from sts:identity_context"
```

Vous pouvez désormais utiliser les informations d'identification reçues à partir de cet appel pour invoquer l'opération d'API `s3:GetDataAccess` et recevoir les informations d'identification finales avec accès à vos ressources S3.

Bien démarrer avec les octrois d'accès S3

Les octrois d'accès Amazon S3 sont une fonctionnalité Amazon S3 qui fournit une solution de contrôle d'accès évolutive pour vos données S3. Les octrois d'accès S3 sont un fournisseur d'informations d'identification S3, ce qui signifie que vous enregistrez avec eux votre liste d'octrois et le niveau d'accès. Par la suite, lorsque des utilisateurs ou des clients ont besoin d'accéder à vos

données S3, ils demandent d'abord leurs informations d'identification aux octrois d'accès S3. S'il existe un octroi correspondant qui autorise l'accès, les octrois d'accès S3 proposent des informations d'identification d'accès temporaires de moindre privilège. Les utilisateurs ou les clients peuvent alors utiliser les informations d'identification proposées par les octrois d'accès S3 pour accéder à vos données S3. Dans cette optique, si vos exigences en matière de données S3 nécessitent une configuration d'autorisations complexe ou importante, vous pouvez utiliser les octrois d'accès S3 pour mettre à l'échelle les autorisations de données S3 pour les utilisateurs, les groupes, les rôles et les applications.

Dans la plupart des cas d'utilisation, vous pouvez gérer le contrôle d'accès à vos données S3 en utilisant AWS Identity and Access Management (IAM) avec des politiques de compartiment ou des politiques basées sur l'identité IAM.

Toutefois, si vous avez des exigences complexes en matière de contrôle d'accès S3, telles que les suivantes, vous pourriez tirer un grand avantage de l'utilisation d'octrois d'accès S3 :

- Vous êtes confronté à la limite de taille de 20 Ko fixée par la politique de compartiment.
- Vous accordez aux identités humaines, telles que les utilisateurs et groupes Microsoft Entra ID (anciennement Azure Active Directory), Okta ou Ping, l'accès aux données S3 pour l'analytique et le big data.
- Vous devez fournir un accès intercompte sans mettre à jour fréquemment les politiques IAM.
- Vos données sont non structurées et de niveau objet au lieu d'être structurées, dans un format de lignes et de colonnes.

Le flux de travail des octrois d'accès S3 est le suivant :

Étapes	Description
1	Création d'une instance d'octrois d'accès S3 Pour commencer, lancez une instance d'octrois d'accès S3 qui contiendra vos octrois d'accès individuels.
2	Enregistrement d'un emplacement Ensuite, enregistrez un emplacement de données S3 (tel que l'emplacement par défaut, `s3://`), puis spécifiez un rôle IAM par défaut que les octrois d'accès S3 endossent lorsqu'ils

Étapes	Description
	fournissent l’accès à l’emplacement de données S3. Vous pouvez également ajouter des emplacements personnalisés à des compartiments ou à des préfixes spécifiques, et les mapper à des rôles IAM personnalisés.
3	Création d’octrois Créez des octrois d’autorisation individuels. Spécifiez dans ces octrois d’autorisation l’emplacement S3 enregistré, la portée de l’accès aux données au sein de cet emplacement, l’identité du bénéficiaire et son niveau d’accès (`READ`, `WRITE` ou `READWRITE`).
4	Demande d’accès aux données S3 Lorsque les utilisateurs, les applications et les utilisateurs Services AWS souhaitent accéder aux données S3, ils font d'abord une demande d'accès. Les octrois d’accès S3 déterminent si la demande doit être autorisée. S’il existe un octroi correspondant qui autorise l’accès, les octrois d’accès S3 utilisent le rôle IAM de l’emplacement enregistré associé à cet octroi pour proposer des informations d’identification temporaires en retour au demandeur.
5	Accédez aux données S3 Les applications utilisent les informations d’identification temporaires proposées par les octrois d’accès S3 pour accéder aux données S3.

Création d’une instance d’octrois d’accès S3

Pour commencer à utiliser les octrois d’accès Amazon S3, vous devez d’abord créer une instance d’octrois d’accès S3. Vous ne pouvez créer qu'une seule instance S3 Access Grants Région AWS par compte. L’instance d’octrois d’accès S3 sert de conteneur pour vos ressources d’octrois d’accès S3, qui incluent les emplacements enregistrés et les octrois.

Avec S3 Access Grants, vous pouvez créer des autorisations pour vos données S3 pour les utilisateurs et les rôles AWS Identity and Access Management (IAM). Si vous avez ajouté votre répertoire d'identité d'entreprise à AWS IAM Identity Center, vous pouvez associer cette instance IAM Identity Center de votre répertoire d'entreprise à votre instance S3 Access Grants. Une fois que vous avez fait cela, vous pouvez créer des octrois d’accès pour les utilisateurs et les groupes de votre entreprise. Si vous n’avez pas encore ajouté votre annuaire d’entreprise à IAM Identity Center, vous pouvez associer ultérieurement votre instance d’octrois d’accès S3 à une instance IAM Identity Center.

Vous pouvez créer une instance S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des kits SDK.

Utilisation de la console S3

Avant de pouvoir accorder l'accès à vos données S3 avec S3 Access Grants, vous devez d'abord créer une instance S3 Access Grants Région AWS identique à vos données S3.

Prérequis

Si vous souhaitez accorder l’accès à vos données S3 en utilisant les identités de votre annuaire d’entreprise, ajoutez votre annuaire d’identités d’entreprise à AWS IAM Identity Center. Si vous n’êtes pas encore prêt à le faire, vous pouvez associer ultérieurement votre instance d’octrois d’accès S3 à une instance IAM Identity Center.

Pour créer une instance d’octrois d’accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation, choisissez le nom du fichier actuellement affiché Région AWS. Ensuite, choisissez la région vers laquelle vous souhaitez passer.
3. Dans le volet de navigation de gauche, choisissez Access Grants.
4. Sur la page Octrois d’accès S3, choisissez Créer une instance d’octrois d’accès S3.
    a. À l’étape 1 de l’Assistant Mise en place de l’instance Access Grants, vérifiez que vous souhaitez créer l’instance dans la Région AWS actuelle. Assurez-vous qu'il s'agit du même Région AWS endroit où se trouvent vos données S3. Vous pouvez créer une instance S3 Access Grants Région AWS par compte.

b. (Facultatif) Si vous avez ajouté votre répertoire d'identité d'entreprise à AWS IAM Identity Center, vous pouvez associer cette instance IAM Identity Center de votre répertoire d'entreprise à votre instance S3 Access Grants.

   Pour ce faire, sélectionnez Ajouter une instance IAM Identity Center dans ***région***. Entrez ensuite l'Amazon Resource Name (ARN) de l'instance IAM Identity Center.

   Si vous n'avez pas encore ajouté votre annuaire d'entreprise à IAM Identity Center, vous pouvez associer ultérieurement votre instance d'octrois d'accès S3 à une instance IAM Identity Center.

c. Pour créer l'instance d'octrois d'accès S3, choisissez Suivant. Pour enregistrer un emplacement, consultez Étape 2 : Enregistrer un emplacement.

5. Si l'option Suivant ou Créer une instance d'octrois d'accès S3 est désactivée :

   Impossible de créer une instance

   - Vous avez peut-être déjà une instance d'octrois d'accès S3 dans la même Région AWS. Dans le volet de navigation de gauche, choisissez Access Grants. Sur la page Octrois d'accès S3, faites défiler la page vers le bas jusqu'à la sectionInstance d'octrois d'accès S3 dans votre compte pour déterminer si une instance existe déjà.
   - Vous n'avez peut-être pas l'autorisation `s3:CreateAccessGrantsInstance` requise pour créer une instance d'octrois d'accès S3. Contactez l'administrateur de votre compte. Pour les autorisations supplémentaires requises si vous associez une instance IAM Identity Center à votre instance d'octrois d'accès S3, consultez CreateAccessGrantsInstance.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example Création d'une instance d'octrois d'accès S3

```
aws s3control create-access-grants-instance \
--account-id 111122223333 \
--region us-east-2
```

Réponse :

```
{
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Utilisation de l'API REST

Vous pouvez utiliser l'API REST Amazon S3 pour créer une instance d'octrois d'accès S3. Pour plus d'informations sur la prise en charge de l'API REST pour la gestion d'une instance d'octrois d'accès S3, consultez les sections suivantes dans la Référence d'API Amazon Simple Storage Service :

- AssociateAccessGrantsIdentityCenter
- CreateAccessGrantsInstance
- DeleteAccessGrantsInstance
- DissociateAccessGrantsIdentityCenter
- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- GetAccessGrantsInstanceResourcePolicy
- ListAccessGrantsInstances
- PutAccessGrantsInstanceResourcePolicy

Utilisation des AWS SDK

Cette section fournit un exemple sur la manière de créer une instance d'octrois d'accès S3 à l'aide des kits AWS SDK.

Java

Cet exemple crée l'instance d'octrois d'accès S3, qui sert de conteneur pour vos octrois d'accès individuels. Vous pouvez avoir une instance S3 Access Région AWS Grants par compte. La réponse inclut l'ID de l'instance `default` et un Amazon Resource Name (ARN) généré pour votre instance d'octrois d'accès S3.

Example Création d'une demande d'instance d'octrois d'accès S3

```
public void createAccessGrantsInstance() {
CreateAccessGrantsInstanceRequest createRequest =
 CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
CreateAccessGrantsInstanceResponse createResponse =
 s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
 " + createResponse);
}
```

Réponse :

```
CreateAccessGrantsInstanceResponse(
CreatedAt=2023-06-07T01:46:20.507Z,
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Rubriques

- Affichage des détails d'une instance d'octrois d'accès S3
- Association ou dissociation de votre instance IAM Identity Center
- Suppression d'une instance d'octrois d'accès S3

Affichage des détails d'une instance d'octrois d'accès S3

Vous pouvez consulter les détails de votre instance d'octrois d'accès Amazon S3 dans une Région AWS particulière. Vous pouvez également répertorier vos instances S3 Access Grants, y compris les instances qui ont été partagées avec vous via AWS Resource Access Manager (AWS RAM).

Vous pouvez consulter les détails de votre instance S3 Access Grants ou répertorier vos instances S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des SDK.

Utilisation de la console S3

Pour consulter une instance d'octrois d'accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d'accès S3, choisissez la région qui contient l'instance d'octrois d'accès S3 qui vous intéresse.
4. La page Octrois d'accès S3 répertorie vos instances d'octrois d'accès S3 et toutes les instances entre comptes qui ont été partagées avec votre compte. Pour visualiser les détails d'une instance, choisissez Afficher les détails.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Obtenir les détails d'une instance d'octrois d'accès S3

```
aws s3control get-access-grants-instance \
 --account-id 111122223333 \
 --region us-east-2
```

Réponse :

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Example : Répertorier toutes les instances d'octrois d'accès S3 pour un compte

Cette action répertorie les instances d'octrois d'accès S3 pour un compte. Vous ne pouvez avoir qu'une seule instance S3 Access Grants par instance Région AWS. Cette action répertorie également les autres instances d'octrois d'accès S3 entre comptes auxquelles votre compte a accès.

```
aws s3control list-access-grants-instances \
 --account-id 111122223333 \
```

```
 --region us-east-2
```

Réponse :

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Utilisation de l'API REST

Pour plus d'informations sur la prise en charge de l'API REST Amazon S3 pour la gestion d'une instance d'octrois d'accès S3, consultez les sections suivantes dans la Référence d'API Amazon Simple Storage Service :

- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- ListAccessGrantsInstances

Utilisation des AWS SDK

Cette section fournit des exemples de la manière d'obtenir les détails d'une instance S3 Access Grants à l'aide AWS des SDK.

Pour utiliser les exemples suivants, remplacez *user input placeholders* par vos propres informations.

Java

Example : Obtenir une instance d'octrois d'accès S3

```
public void getAccessGrantsInstance() {
GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
GetAccessGrantsInstanceResponse getResponse =
 s3Control.getAccessGrantsInstance(getRequest);
LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
```

```
}
```

Réponse :

```
GetAccessGrantsInstanceResponse(
AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
CreatedAt=2023-06-07T01:46:20.507Z)
```

Example : Répertorier toutes les instances d'octrois d'accès S3 pour un compte

Cette action répertorie les instances d'octrois d'accès S3 pour un compte. Vous ne pouvez disposer que d'une seule instance d'octrois d'accès S3 par région. Cette action peut également répertorier d'autres instances d'octrois d'accès S3 entre comptes auxquelles votre compte a accès.

```
public void listAccessGrantsInstances() {
ListAccessGrantsInstancesRequest listRequest =
 ListAccessGrantsInstancesRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsInstancesResponse listResponse =
 s3Control.listAccessGrantsInstances(listRequest);
LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Réponse :

```
ListAccessGrantsInstancesResponse(
AccessGrantsInstancesList=[
ListAccessGrantsInstanceEntry(
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
CreatedAt=2023-06-07T04:28:11.728Z
)
]
)
```

Association ou dissociation de votre instance IAM Identity Center

Dans Amazon S3 Access Grants, vous pouvez associer l' AWS IAM Identity Center instance de votre répertoire d'identité d'entreprise à une instance S3 Access Grants. Ensuite, vous pouvez créer des

autorisations d'accès pour les utilisateurs et les groupes de votre annuaire d'entreprise, en plus des utilisateurs et des rôles AWS Identity and Access Management (IAM).

Si vous ne souhaitez plus créer d'octrois d'accès pour les utilisateurs et les groupes de votre annuaire d'entreprise, vous pouvez dissocier votre instance IAM Identity Center de votre instance d'octrois d'accès S3.

Vous pouvez associer ou dissocier une instance IAM Identity Center à l'aide de la console Amazon S3, de l' AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 ou des kits AWS SDK.

Utilisation de la console S3

Avant d'associer votre instance IAM Identity Center à votre instance d'octrois d'accès S3, vous devez ajouter votre annuaire d'identités d'entreprise à IAM Identity Center. Pour plus d'informations, consultez the section called "Octrois d'accès S3 et identités d'annuaire d'entreprise".

Pour associer une instance IAM Identity Center à une instance d'octrois d'accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d'accès S3, choisissez la région qui contient l'instance d'octrois d'accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page de détails, dans la section IAM Identity Center, choisissez Ajouter pour ajouter une instance IAM Identity Center ou Désenregistrer pour annuler l'enregistrement d'une instance IAM Identity Center déjà associée.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Associer une instance IAM Identity Center à une instance d'octrois d'accès S3

```
aws s3control associate-access-grants-identity-center \
```

```
 --account-id 111122223333 \
 --identity-center-arn arn:aws:sso:::instance/ssoins-1234a567bb89012c \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Example : Dissocier une instance IAM Identity Center d'une instance d'octrois d'accès S3

```
aws s3control dissociate-access-grants-identity-center \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l'API REST Amazon S3 pour la gestion de l'association entre une instance IAM Identity Center et une instance d'octrois d'accès S3, consultez les sections suivantes dans la Référence d'API Amazon Simple Storage Service :

- AssociateAccessGrantsIdentityCenter
- DissociateAccessGrantsIdentityCenter

Suppression d'une instance d'octrois d'accès S3

Vous pouvez supprimer une instance Amazon S3 Access Grants depuis un Région AWS compte. Toutefois, avant de supprimer une instance d'octrois d'accès S3, vous devez effectuer les opérations suivantes :

- Supprimez toutes les ressources figurant dans l'instance d'octrois d'accès S3, y compris tous les octrois et tous les emplacements. Pour plus d'informations, consultez Suppression d'un octroi et Suppression d'un emplacement.
- Si vous avez associé une AWS IAM Identity Center instance à votre instance S3 Access Grants, vous devez dissocier l'instance IAM Identity Center. Pour plus d'informations, consultez Association ou dissociation de votre instance IAM Identity Center.

> **Important**
>
> Si vous supprimez une instance d’octrois d’accès S3, la suppression est permanente et ne peut pas être annulée. Tous les bénéficiaires auxquels l’accès a été accordé via les octrois figurant dans cette instance d’octrois d’accès S3 perdront l’accès à vos données S3.

Vous pouvez supprimer une instance S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des kits SDK.

Utilisation de la console S3

Pour supprimer une instance d’octrois d’accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d’accès S3, choisissez la région qui contient l’instance d’octrois d’accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page de détails de l’instance, choisissez Supprimer une instance dans le coin supérieur droit.
6. Dans la boîte de dialogue qui s’affiche, choisissez Supprimer. Cette action ne peut pas être annulée.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *`user input placeholders`* par vos propres informations.

> **Note**
>
> Avant de pouvoir supprimer une instance d’octrois d’accès S3, vous devez d’abord supprimer tous les octrois et tous les emplacements créés dans l’instance d’octrois d’accès S3. Si vous

avez associé une instance du centre IAM Identity Center à votre instance d'octrois d'accès S3, vous devez d'abord l'en dissocier.

Example : Supprimer une instance d'octrois d'accès S3

```
aws s3control delete-access-grants-instance \
--account-id 111122223333 \
--profile access-grants-profile \
--region us-east-2 \
--endpoint-url https://s3-control.us-east-2.amazonaws.com \

 // No response body
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l'API REST Amazon S3 pour la suppression d'une instance d'octrois d'accès S3, consultez DeleteAccessGrantsInstance dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS SDK

Cette section fournit des exemples illustrant la manière de supprimer une instance d'octrois d'accès S3 à l'aide des kits AWS SDK.

Pour utiliser l'exemple suivant, remplacez *user input placeholders* par vos propres informations.

Java

> **Note**
>
> Avant de pouvoir supprimer une instance d'octrois d'accès S3, vous devez d'abord supprimer tous les octrois et tous les emplacements créés dans l'instance d'octrois d'accès S3. Si vous avez associé une instance du centre IAM Identity Center à votre instance d'octrois d'accès S3, vous devez d'abord l'en dissocier.

Example : Supprimer une instance d'octrois d'accès S3

```
public void deleteAccessGrantsInstance() {
```

```
DeleteAccessGrantsInstanceRequest deleteRequest =
 DeleteAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
DeleteAccessGrantsInstanceResponse deleteResponse =
 s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Enregistrement d’un emplacement

Après avoir créé une instance Amazon S3 Access Grants Région AWS dans un emplacement de votre compte, vous pouvez enregistrer un emplacement S3 dans cette instance. Un emplacement est une ressource S3 qui contient des données auxquelles vous souhaitez accorder l’accès. Vous pouvez enregistrer l'emplacement par défaut`s3://`, c'est-à-dire tous vos compartiments Région AWS, puis réduire l'étendue de l'accès ultérieurement, lorsque vous créez des autorisations d'accès individuelles. Vous pouvez également enregistrer un compartiment spécifique ou un compartiment et un préfixer comme emplacement.

Vous devez d’abord enregistrer au moins un emplacement auprès de votre instance d’octrois d’accès S3 avant de pouvoir créer des octrois d’accès. Lorsque vous enregistrez un emplacement, vous devez également spécifier le rôle AWS Identity and Access Management (IAM) que les octrois d’accès S3 endossent pour répondre aux demandes d’exécution relatives à l’emplacement et étendre les autorisations jusqu’à l’octroi spécifique au moment de l’exécution.

URI S3	Rôle IAM	Description
`s3://`	*`Default-IAM-role`*	L’emplacement par défaut, `s3://`, inclut tous les compartiments figurant dans la Région AWS.
`s3://`*`example-s3-bucket1`*`/`	*`IAM-role-For-bucket`*	Cet emplacement inclut tous les objets figurant dans le compartiment spécifié.

Avant de pouvoir enregistrer un emplacement, assurez-vous d’effectuer les opérations suivantes :

- Créez un ou plusieurs compartiments contenant les données auxquelles vous souhaitez accorder l'accès. Ces compartiments doivent être situés au même endroit Région AWS que votre instance S3 Access Grants. Pour plus d'informations, consultez Création d'un compartiment.

  Pour ajouter un préfixe à un compartiment, consultez Création de noms de clés d'objets.

- Créez un rôle IAM et accordez au principal du service d'octrois d'accès S3 l'accès à ce rôle dans le fichier de politique de ressources. Pour ce faire, vous pouvez créer un fichier JSON contenant les instructions suivantes. Pour ajouter la politique de ressources à votre compte, consultez Créer et attacher votre première politique gérée par le client.

  TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity", "sts:SetContext"],
      "Effect": "Allow",
      "Principal": {"Service":"access-grants.s3.amazonaws.com"}
    }
  ]
}
```

- Créez une politique IAM pour attacher les autorisations Amazon S3 au rôle IAM. Consultez l'exemple de fichier `iam-policy.json` suivant et remplacez les *`user input placeholders`* par vos propres informations.

  > **Note**
  >
  > - Si vous utilisez le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) pour chiffrer vos données, l'exemple suivant inclut les AWS KMS autorisations nécessaires pour le rôle IAM dans la politique. Si vous n'utilisez pas cette fonctionnalité, vous pouvez supprimer ces autorisations de votre politique IAM.
  > - Vous pouvez restreindre le rôle IAM pour accéder aux données S3 uniquement si les informations d'identification sont fournies par S3 Access Grants. Cet exemple vous montre comment ajouter une `Condition` instruction pour une instance S3 Access Grants spécifique. Pour ce faire, remplacez l'ARN de l'instance S3 Access Grants dans

l'instruction de condition par l'ARN de votre instance S3 Access Grants, au format suivant : `arn:aws:s3:region:accountId:access-grants/default`

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
```

```
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Région
 AWS:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Région
 AWS:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Vous pouvez enregistrer un emplacement dans votre instance S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 ou AWS des SDK.

Utilisation de la console S3

Avant de pouvoir accorder l’accès à vos données S3 avec les octrois d’accès S3, vous devez avoir au moins un emplacement enregistré.

Pour enregistrer un emplacement dans votre instance d’octrois d’accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d’accès S3, choisissez la région qui contient l’instance d’octrois d’accès S3 qui vous intéresse.

   Si vous utilisez une instance d’octrois d’accès S3 pour la première fois, assurez-vous d’avoir terminé l’étape 1 : Créer une instance d’octrois d’accès S3 et d’être passé à l’étape 2 de l’Assistant Mise en place de l’instance Access Grants. Si vous possédez déjà une instance d’octrois d’accès S3, choisissez Afficher les détails, puis, dans l’onglet Succursales, choisissez Enregistrer l’emplacement.

   a. Pour Champ d’application de l’emplacement, choisissez Parcourir S3 ou entrez le chemin d’URI S3 menant à l’emplacement que vous souhaitez enregistrer. Pour les formats d’URI S3, consultez le tableau des formats d’emplacement. Après avoir entré un URI, vous pouvez choisir Afficher pour accéder à l’emplacement.

   b. Pour Rôle IAM choisissez l’une des options suivantes :

      - Sélectionner parmi les rôles IAM existants

        Choisissez un rôle IAM dans la liste déroulante. Après avoir choisi un rôle, choisissez Afficher pour vous assurer que ce rôle dispose des autorisations nécessaires pour gérer l’emplacement que vous enregistrez. Plus précisément, assurez-vous que ce rôle accorde aux octrois d’accès S3 les autorisations `sts:AssumeRole` et `sts:SetSourceIdentity`.

      - Saisir l’ARN du rôle IAM

Accédez à la console IAM. Copiez le nom de ressource Amazon (ARN) du rôle IAM et collez-le dans cette zone.

c. Pour terminer, choisissez Suivant ou Enregistrer l'emplacement.

4. Résolution de problèmes

Impossible d'enregistrer l'emplacement

- L'emplacement est peut-être déjà enregistré.

  Il se peut que vous n'ayez pas l'autorisation `s3:CreateAccessGrantsLocation` d'enregistrer des emplacements. Contactez l'administrateur de votre compte.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Vous pouvez enregistrer l'emplacement par défaut, `s3://`, ou un emplacement personnalisé dans votre instance d'octrois d'accès S3. Assurez-vous d'abord de créer un rôle IAM avec un accès principal à l'emplacement, puis d'accorder aux octrois d'accès S3 l'autorisation d'endosser ce rôle.

Pour utiliser les exemples de commandes suivants, remplacez les *`user input placeholders`* par vos propres informations.

Example Créer une politique de ressources

Créez une politique permettant aux octrois d'accès S3 d'endosser le rôle IAM. Pour ce faire, vous pouvez créer un fichier JSON contenant les instructions suivantes. Pour ajouter la politique de ressources à votre compte, consultez Créer et attacher votre première politique gérée par le client.

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
```

```
      "Effect": "Allow",
      "Principal": {"Service":"access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Créer le rôle

Exécutez la commande IAM suivante pour créer le rôle.

```
aws iam create-role --role-name accessGrantsTestRole \
 --region us-east-2 \
 --assume-role-policy-document file://TestRolePolicy.json
```

L'exécution de la commande `create-role` renvoie la politique :

```
{
    "Role": {
        "Path": "/",
        "RoleName": "accessGrantsTestRole",
        "RoleId": "AROASRDGX4WM4GH55GIDA",
        "Arn": "arn:aws:iam::111122223333:role/accessGrantsTestRole",
        "CreateDate": "2023-05-31T18:11:06+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Stmt1685556427189",
                    "Action": [
                        "sts:AssumeRole",
                        "sts:SetSourceIdentity"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service":"access-grants.s3.amazonaws.com"
                    }
                }
            ]
        }
    }
}
```

Example

Créez une politique IAM pour attacher les autorisations Amazon S3 au rôle IAM. Consultez l'exemple de fichier `iam-policy.json` suivant et remplacez les *`user input placeholders`* par vos propres informations.

> **Note**
>
> Si vous utilisez le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) pour chiffrer vos données, l'exemple suivant ajoute les AWS KMS autorisations nécessaires pour le rôle IAM dans la politique. Si vous n'utilisez pas cette fonctionnalité, vous pouvez supprimer ces autorisations de votre politique IAM.
>
> Pour vous assurer que le rôle IAM ne peut être utilisé que pour accéder aux données dans S3 si les informations d'identification sont proposées par les octrois d'accès S3, cet exemple vous montre comment ajouter une déclaration `Condition` spécifiant l'instance d'octrois d'accès S3 (`s3:AccessGrantsInstance:` *`InstanceArn`*) dans votre politique IAM. Lorsque vous utilisez l'exemple de politique suivant, remplacez les *`user input placeholders`* par vos propres informations.

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
```

```
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Région AWS:accountId:access-
grants/default"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:Région AWS:accountId:access-
grants/default"]
            }
         }
```

```
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Example

Exécutez la commande suivante :

```
aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json
```

Example Enregistrer l'emplacement par défaut

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Réponse :

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example Enregistrer un emplacement personnalisé

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3://DOC-BUCKET-EXAMPLE/ \
 --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Réponse :

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-BUCKET-EXAMPLE/",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Utilisation de l'API REST

Pour plus d’informations sur la prise en charge de l’API REST Amazon S3 pour la gestion d’une instance d’octrois d’accès S3, consultez les sections suivantes dans la Référence d’API Amazon Simple Storage Service :

- CreateAccessGrantsLocation
- DeleteAccessGrantsLocation
- GetAccessGrantsLocation
- ListAccessGrantsLocations
- UpdateAccessGrantsLocation

Utilisation des AWS SDK

Cette section fournit des exemples illustrant la manière d’enregistrer des emplacements à l’aide des kits AWS SDK.

Pour utiliser les exemples suivants, remplacez *user input placeholders* par vos propres informations.

Java

Vous pouvez enregistrer l'emplacement par défaut, `s3://`, ou un emplacement personnalisé dans votre instance d'octrois d'accès S3. Assurez-vous d'abord de créer un rôle IAM avec un accès principal à l'emplacement, puis d'accorder aux octrois d'accès S3 l'autorisation d'endosser ce rôle.

Pour utiliser les exemples de commandes suivants, remplacez les *`user input placeholders`* par vos propres informations.

Example Enregistrer un emplacement par défaut

Requête :

```
public void createAccessGrantsLocation() {
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://")
.iamRoleArn("arn:aws:iam::123456789012:role/accessGrantsTestRole")
.build();
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Réponse :

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example Enregistrer un emplacement personnalisé

Requête :

```
public void createAccessGrantsLocation() {
```

```
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://DOC-BUCKET-EXAMPLE/")
.iamRoleArn("arn:aws:iam::111122223333:role/accessGrantsTestRole")
.build();
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Réponse :

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=18cfe6fb-eb5a-4ac5-aba9-8d79f04c2012,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/18cfe6fb-eb5a-4ac5-aba9-8d79f04c2666,
LocationScope= s3://test-bucket-access-grants-user123/,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Rubriques

- Affichage des détails d’un emplacement enregistré
- Mise à jour d’un emplacement enregistré
- Suppression d’un emplacement enregistré

Affichage des détails d’un emplacement enregistré

Vous pouvez obtenir les détails d'un emplacement enregistré dans votre instance S3 Access Grants en utilisant la console Amazon S3, le AWS Command Line Interface (AWS CLI), l'API REST Amazon S3 et les AWS SDK.

Utilisation de la console S3

Pour afficher les emplacements enregistrés dans votre instance d’octrois d’accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d'accès S3, choisissez la région qui contient l'instance d'octrois d'accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page de détails de l'instance, choisissez l'onglet Succursales.
6. Recherchez l'emplacement enregistré que vous souhaitez afficher. Pour filtrer la liste des emplacements enregistrés, utilisez la zone de recherche.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Obtenir les détails d'un emplacement enregistré

```
aws s3control get-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id default
```

Réponse :

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example : Répertorier tous les emplacements enregistrés dans une instance d'octrois d'accès S3

Pour limiter les résultats à un préfixe ou à un compartiment S3, vous pouvez éventuellement utiliser le paramètre `--location-scope s3://bucket-and-or-prefix`.

```
aws s3control list-access-grants-locations \
```

```
--account-id 111122223333 \
--region us-east-2
```

Réponse :

```
{"AccessGrantsLocationsList": [
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     },
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     }
   ]
  }
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l’API REST Amazon S3 pour obtenir les détails d’un emplacement enregistré ou répertorier tous les emplacements enregistrés auprès d’une instance d’octrois d’accès S3, consultez les sections suivantes de la Référence d’API Amazon Simple Storage Service :

- GetAccessGrantsLocation
- ListAccessGrantsLocations

Utilisation des AWS SDK

Cette section fournit des exemples de la manière d’obtenir les détails d’un emplacement enregistré ou de répertorier tous les emplacements enregistrés dans une instance d’octrois d’accès S3 à l’aide des kits AWS SDK.

Pour utiliser les exemples suivants, remplacez *user input placeholders* par vos propres informations.

Java

Example : Obtenir les détails d'un emplacement enregistré

```
public void getAccessGrantsLocation() {
GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
 GetAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("default")
.build();
GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
 s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Réponse :

```
GetAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope= s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example : Répertorier tous les emplacements enregistrés dans une instance d'octrois d'accès S3

Pour limiter les résultats à un préfixe ou à un compartiment S3, vous pouvez éventuellement transmettre un URI S3, tel que `s3://bucket-and-or-prefix`, dans le paramètre `LocationScope`.

```
public void listAccessGrantsLocations() {

ListAccessGrantsLocationsRequest listRequest =
 ListAccessGrantsLocationsRequest.builder()
.accountId("111122223333")
.build();
```

```
ListAccessGrantsLocationsResponse listResponse =
 s3Control.listAccessGrantsLocations(listRequest);
LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Réponse :

```
ListAccessGrantsLocationsResponse(
AccessGrantsLocationsList=[
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
),
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixA*,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
]
)
```

Mise à jour d’un emplacement enregistré

Vous pouvez mettre à jour le rôle AWS Identity and Access Management (IAM) d'un emplacement enregistré dans votre instance Amazon S3 Access Grants. Pour chaque nouveau rôle IAM que vous utilisez pour enregistrer un emplacement dans les octrois d’accès S3, veillez à accorder au principal du service d’octrois d’accès S3 (`access-grants.s3.amazonaws.com`) l’accès à ce rôle. Pour ce faire, ajoutez une entrée pour le nouveau rôle IAM dans le même fichier JSON de politique d’approbation que celui que vous avez utilisé quand vous avez enregistré l’emplacement pour la première fois.

Vous pouvez mettre à jour un emplacement dans votre instance S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des SDK.

Utilisation de la console S3

Pour mettre à jour le rôle IAM d'un emplacement enregistré auprès de votre instance d'octrois d'accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d'accès S3, choisissez la région qui contient l'instance d'octrois d'accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page de détails de l'instance, choisissez l'onglet Succursales.
6. Recherchez l'emplacement que vous souhaitez mettre à jour. Pour filtrer la liste des emplacements, utilisez la zone de recherche.
7. Choisissez le bouton d'options en regard de l'emplacement enregistré que vous voulez mettre à jour.
8. Mettez à jour le rôle IAM, puis choisissez Enregistrer les modifications.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Mettre à jour le rôle IAM d'un emplacement enregistré

```
aws s3control update-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id 635f1139-1af2-4e43-8131-a4de006eb999 \
--iam-role-arn arn:aws:iam::777788889999:role/accessGrantsTestRole
```

Réponse :

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
```

```
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*",
    "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l'API REST Amazon S3 pour la mise à jour d'un emplacement dans une instance d'octrois d'accès S3, consultez UpdateAccessGrantsLocation dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS SDK

Cette section fournit des exemples de mise à jour du rôle IAM d'un emplacement enregistré à l'aide des AWS SDK.

Pour utiliser l'exemple suivant, remplacez *user input placeholders* par vos propres informations.

Java

Example : Mettre à jour le rôle IAM d'un emplacement enregistré

```
public void updateAccessGrantsLocation() {
UpdateAccessGrantsLocationRequest updateRequest =
 UpdateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("635f1139-1af2-4e43-8131-a4de006eb999")
.iamRoleArn("arn:aws:iam::777788889999:role/accessGrantsTestRole")
.build();
UpdateAccessGrantsLocationResponse updateResponse =
 s3Control.updateAccessGrantsLocation(updateRequest);
LOGGER.info("UpdateAccessGrantsLocationResponse: " + updateResponse);
}
```

Réponse :

```
UpdateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,
```

```
AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://DOC-EXAMPLE-BUCKET/prefixB*,
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Suppression d’un emplacement enregistré

Vous pouvez supprimer un enregistrement d’emplacement à partir d’une instance d’octrois d’accès Amazon S3. La suppression de l’emplacement désenregistre ce dernier de l’instance d’octrois d’accès S3.

Avant de pouvoir supprimer un enregistrement d’emplacement à partir d’une instance d’octrois d’accès S3, vous devez supprimer tous les octrois associés à cet emplacement. Pour en savoir plus sur la suppression d’octrois, consultez Suppression d’un octroi.

Vous pouvez supprimer un emplacement dans votre instance S3 Access Grants à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des SDK.

Utilisation de la console S3

Pour supprimer un enregistrement d’emplacement de votre instance d’octrois d’accès S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d’accès S3, choisissez la région qui contient l’instance d’octrois d’accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page de détails de l’instance, choisissez l’onglet Succursales.
6. Recherchez l’emplacement que vous souhaitez mettre à jour. Pour filtrer la liste des emplacements, utilisez la zone de recherche.
7. Choisissez le bouton d’option en regard de l’emplacement enregistré que vous voulez supprimer.
8. Choisissez Deregister (Annuler l'enregistrement).
9. Une boîte de dialogue apparaît pour vous avertir que cette action ne peut pas être annulée. Pour supprimer l’emplacement, choisissez Désenregistrer.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Supprimer un enregistrement d’emplacement

```
aws s3control delete-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 // No response body
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l’API REST Amazon S3 pour la suppression d’un emplacement à partir d’une instance d’octrois d’accès S3, consultez DeleteAccessGrantsLocation dans la Référence d’API Amazon Simple Storage Service.

Utilisation des AWS SDK

Cette section fournit un exemple illustrant la manière de supprimer un emplacement à l’aide des kits AWS SDK.

Pour utiliser l'exemple suivant, remplacez *user input placeholders* par vos propres informations.

Java

Example : Supprimer un enregistrement d’emplacement

```
public void deleteAccessGrantsLocation() {
DeleteAccessGrantsLocationRequest deleteRequest =
 DeleteAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantsLocationResponse deleteResponse =
 s3Control.deleteAccessGrantsLocation(deleteRequest);
LOGGER.info("DeleteAccessGrantsLocationResponse: " + deleteResponse);
```

```
}
```

Réponse :

```
DeleteAccessGrantsLocationResponse()
```

Création d'octrois

Après avoir enregistré au moins un emplacement dans votre instance d'octrois d'accès Amazon S3, vous pouvez créer un octroi d'accès. Un octroi d'accès donne au bénéficiaire l'autorisation d'accéder à un emplacement enregistré.

Le bénéficiaire peut être un utilisateur ou un rôle AWS Identity and Access Management (IAM) ou un utilisateur ou un groupe d'annuaires. Un utilisateur d'annuaire est un utilisateur issu de votre annuaire d'entreprise ou d'une source d'identité externe que vous avez ajouté à l'instance AWS IAM Identity Center qui est associée à votre instance d'octrois d'accès S3. Pour créer un octroi pour un utilisateur ou un groupe spécifique à partir d'IAM Identity Center, recherchez le GUID utilisé par IAM Identity Center pour identifier cet utilisateur dans IAM Identity Center, par exemple, `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`.

Vous pouvez accorder l'accès à un compartiment, à un préfixe ou à un objet. Dans Amazon S3, un préfixe est une chaîne de caractères située au début d'un nom de clé d'objet et qui sert à organiser les objets au sein d'un compartiment. Il peut s'agir de n'importe quelle chaîne de caractères autorisés, par exemple des noms de clé d'objet d'un compartiment commençant par le préfixe `engineering/`.

Sous-préfixe

Lorsque vous accordez l'accès à un emplacement enregistré, vous pouvez utiliser le champ `Subprefix` pour restreindre la portée à un préfixe spécifique dans un compartiment ou à un objet spécifique dans un compartiment.

Vous ne pouvez pas créer d'octroi d'accès pour l'emplacement par défaut `s3://`, ce qui permettrait au bénéficiaire d'accéder à chaque compartiment d'une région. Si vous choisissez l'emplacement `s3://` par défaut comme emplacement d'octroi, vous devez réduire la portée de l'octroi en utilisant le champ `Subprefix` pour spécifier l'un des éléments suivants :

- Un compartiment : `s3://bucket/*`

- Un préfixe dans un compartiment : `s3://bucket/prefix*`
- Un préfixe dans un préfixe : `s3://bucket/prefixA/prefixB*`
- Un objet : `s3://bucket/object-key-name`

Si vous créez un octroi d'accès où l'emplacement enregistré est un compartiment, vous pouvez transmettre l'un des éléments suivants dans le champ `Subprefix` :

- Un préfixe dans le compartiment : *prefix**
- Un préfixe dans un préfixe : *prefixA/prefixB**
- Un objet : */object-key-name*

L'étendue de l'autorisation affichée dans la console Amazon S3 ou `GrantScope` celle renvoyée dans la réponse API ou AWS Command Line Interface (AWS CLI) est le résultat de la concaténation du chemin de localisation avec le. `Subprefix` Assurez-vous que ce chemin concaténé correspond bien au compartiment, au préfixe ou à l'objet S3 auquel vous souhaitez accorder l'accès.

Si vous créez un octroi d'accès qui accorde l'accès à un seul objet, spécifiez dans l'appel d'API ou dans la commande CLI que le type `s3PrefixType` est `Object`.

> **Note**
>
> Vous ne pouvez pas créer d'octroi à un compartiment si celui-ci n'existe pas encore. Toutefois, vous pouvez créer un octroi à un préfixe qui n'existe pas encore.

Vous pouvez créer une autorisation d'accès à l'aide de la console Amazon S3 AWS CLI, de l'API REST Amazon S3 et des AWS kits SDK.

Utilisation de la console S3

Pour créer un octroi d'accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d'accès S3, choisissez la région qui contient l'instance d'octrois d'accès S3 qui vous intéresse.

Si vous utilisez l'instance d'octrois d'accès S3 pour la première fois, assurez-vous d'avoir terminé l'étape 2 : Enregistrer un emplacement et d'être passé à l'étape 3 de l'Assistant Mise en place de l'instance Access Grants. Si vous possédez déjà une instance d'octrois d'accès S3, choisissez Afficher les détails, puis, dans l'onglet Octrois, choisissez Créer un octroi.

a. Dans la section Portée de l'octroi, sélectionnez ou entrez un emplacement enregistré.

   Si vous avez sélectionné l'emplacement `s3://` par défaut, utilisez la zone Sous-préfixe pour réduire la portée de l'octroi d'accès. Pour plus d'informations, consultez Sous-préfixe. Si vous accordez l'accès uniquement à un objet, sélectionnez Le périmètre de l'octroi est un objet.

b. Sous Autorisations et accès, sélectionnez le niveau d'autorisation : Lecture, Écriture ou les deux.

   Choisissez ensuite le type de bénéficiaire. Si vous avez ajouté votre annuaire d'entreprise à IAM Identity Center et associé cette instance IAM Identity Center à votre instance d'octrois d'accès S3, vous pouvez choisir Identité du répertoire depuis IAM Identity Center. Si vous choisissez cette option, obtenez l'ID de l'utilisateur ou du groupe auprès d'IAM Identity Center et entrez-le dans cette section.

   Si le type de bénéficiaire est un utilisateur ou un rôle IAM, choisissez Principal IAM. Sous Type principal d'IAM, choisissez Utilisateur ou Rôle. Ensuite, sous Utilisateur principal d'IAM, choisissez dans la liste ou entrez l'ID de l'identité.

c. Pour créer l'octroi d'accès S3, choisissez Suivant ou Créer un octroi.

4. Si l'option Suivant ou Créer un octroi est désactivée :

   Impossible de créer un octroi

   - Il se peut que vous deviez d'abord enregistrer un emplacement dans votre instance d'octrois d'accès S3.
   - Vous ne disposez peut-être pas de l'autorisation `s3:CreateAccessGrant` pour créer un octroi d'accès. Contactez l'administrateur de votre compte.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Les exemples suivants montrent comment créer une demande d'octroi d'accès pour un principal IAM et comment créer une demande d'octroi d'accès pour un utilisateur ou un groupe d'annuaire d'entreprise.

Pour utiliser les exemples de commandes suivants, remplacez les *user input placeholders* par vos propres informations.

> **Note**
>
> Si vous créez un octroi d'accès qui accorde l'accès à un seul objet, incluez le paramètre requis `--s3-prefix-type Object`.

Example Créer une demande d'octroi d'accès pour un principal IAM

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
--access-grants-location-configuration S3SubPrefix=prefixB* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-consumer-3
```

Example Créer une réponse d'octroi d'accès

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/grant/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*",
    "Permission": "READ"
}
```

Créer une demande d’octroi d’accès pour un utilisateur ou un groupe d’annuaire

Pour créer une demande d’octroi d’accès pour un utilisateur ou un groupe d’annuaire, vous devez d’abord obtenir le GUID de l’utilisateur ou du groupe d’annuaire en exécutant l’une des commandes suivantes.

Example Obtenir un GUID pour un utilisateur ou un groupe d’annuaire

Vous pouvez trouver le GUID d'un utilisateur d'IAM Identity Center via la console IAM Identity Center ou à l'aide des kits de développement logiciel (SDK). AWS CLI AWS La commande suivante répertorie les utilisateurs de l’instance IAM Identity Center spécifiée, avec leurs noms et identifiants.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Cette commande répertorie les groupes figurant dans l’instance IAM Identity Center spécifiée.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Créer un octroi d’accès pour un utilisateur ou un groupe d’annuaire

Cette commande est similaire à la création d’un octroi pour des utilisateurs ou des rôles IAM, si ce n’est que le type de bénéficiaire est `DIRECTORY_USER` ou `DIRECTORY_GROUP`, et que l’identifiant du bénéficiaire est le GUID de l’utilisateur ou du groupe d’annuaire.

```
aws s3control create-access-grant \
--account-id 123456789012 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix="DOC-EXAMPLE-BUCKET/rafael/*" \
--permission READWRITE \
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-
f1d683aacba5 \
```

Utilisation de l'API REST

Pour plus d’informations sur la prise en charge de l’API REST Amazon S3 pour la gestion des octrois d’accès, consultez les sections suivantes dans la Référence d’API Amazon Simple Storage Service :

- CreateAccessGrant
- DeleteAccessGrant

- GetAccessGrant
- ListAccessGrants

Utilisation des AWS SDK

Cette section fournit des exemples illustrant la manière de créer un octroi d'accès à l'aide des kits AWS SDK.

Java

Pour utiliser l'exemple suivant, remplacez les *`user input placeholders`* par vos propres informations :

> **Note**
>
> Si vous créez un octroi d'accès qui accorde l'accès à un seul objet, incluez le paramètre requis `.s3PrefixType(S3PrefixType.Object)`.

Example Créer une demande d'octroi d'accès

```
public void createAccessGrant() {
CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa")
.permission("READ")
.accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPrefix("
.grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223333:u
data-consumer-3").build())
.build();
CreateAccessGrantResponse createResponse =
 s3Control.createAccessGrant(createRequest);
LOGGER.info("CreateAccessGrantResponse: " + createResponse);
}
```

Example Créer une réponse d'octroi d'accès

```
CreateAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
```

```
AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://DOC-BUCKET-EXAMPLE/prefixB,
Permission=READ
)
```

Rubriques

Affichage d’un octroi

Vous pouvez consulter les détails d'une autorisation d'accès dans votre instance Amazon S3 Access Grants en utilisant la console Amazon S3, le AWS Command Line Interface (AWS CLI), l'API REST Amazon S3 et les AWS SDK.

Utilisation de la console S3

Pour afficher les détails d’un octroi d’accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d’accès S3, choisissez la région qui contient l’instance d’octrois d’accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page des détails, choisissez l’onglet Octrois.
6. Dans la section Octrois, recherchez l’octroi d’accès que vous souhaitez consulter. Pour filtrer la liste des octrois, utilisez la zone de recherche.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser les exemples de commandes suivants, remplacez les *user input placeholders* par vos propres informations.

Example : Obtenir les détails d'un octroi d'accès

```
aws s3control get-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Réponse :

```
{
    "CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://DOC-EXAMPLE-BUCKET/"
}
```

Example : Répertorier tous les octrois d'accès dans une instance d'octrois d'accès S3

Vous pouvez éventuellement utiliser les paramètres suivants pour limiter les résultats à un préfixe S3 ou à une identité AWS Identity and Access Management (IAM) :

- Sous-préfixe : `--grant-scope s3://bucket-name/prefix*`
- Identité IAM : `--grantee-type IAM` et `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Réponse :

```
{
    "AccessGrantsList": [{"CreatedAt": "2023-06-14T17:54:46.542000+00:00",
            "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcarda1",
            "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixA*"
        },
        {"CreatedAt": "2023-06-24T17:54:46.542000+00:00",
            "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
            "GrantScope": "s3://DOC-EXAMPLE-BUCKET/prefixB*"
        },

    ]
}
```

Utilisation de l'API REST

Vous pouvez utiliser les opérations d’API Amazon S3 pour consulter les détails d’un octroi d’accès et répertorier tous les octrois d’accès dans une instance d’octrois d’accès S3. Pour en savoir plus sur la prise en charge de l’API REST pour la gestion des octrois d’accès, consultez les sections suivantes dans la Référence d’API Amazon Simple Storage Service :

- GetAccessGrant

- ListAccessGrants

Utilisation des AWS SDK

Cette section fournit des exemples de la manière d'obtenir les détails d'une autorisation d'accès à l'aide AWS des SDK.

Pour utiliser les exemples suivants, remplacez *user input placeholders* par vos propres informations.

Java

Example : Obtenir les détails d'un octroi d'accès

```
public void getAccessGrant() {
GetAccessGrantRequest getRequest = GetAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE22222")
.build();
GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);
LOGGER.info("GetAccessGrantResponse: " + getResponse);
}
```

Réponse :

```
GetAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://DOC-EXAMPLE-BUCKET/
)
```

Example : Répertorier tous les octrois d'accès dans une instance d'octrois d'accès S3

Vous pouvez éventuellement utiliser ces paramètres pour limiter les résultats à un préfixe S3 ou à une identité IAM :

- Portée : `GrantScope=s3://bucket-name/prefix*`
- Bénéficiaire : `GranteeType=IAM` et `GranteeIdentifier= arn:aws:iam::111122223333:role/accessGrantsConsumerRole`

```
public void listAccessGrants() {
ListAccessGrantsRequest listRequest = ListAccessGrantsRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsResponse listResponse = s3Control.listAccessGrants(listRequest);
LOGGER.info("ListAccessGrantsResponse: " + listResponse);
}
```

Réponse :

```
ListAccessGrantsResponse(
AccessGrantsList=[
ListAccessGrantEntry(
CreatedAt=2023-06-14T17:54:46.540z,
AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcarda1,
GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixA
),
ListAccessGrantEntry(
CreatedAt=2023-06-24T17:54:46.540Z,
AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
```

```
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
),
Permission=READ,
AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
GrantScope=s3://DOC-EXAMPLE-BUCKET/prefixB*
)
]
)
```

Suppression d’un octroi

Vous pouvez supprimer des octrois d’accès de votre instance d’octrois d’accès Amazon S3. Vous ne pouvez pas annuler la suppression d’un octroi d’accès. Une fois que vous avez supprimé un octroi d’accès, le bénéficiaire n’a plus accès à vos données Amazon S3.

Vous pouvez supprimer une autorisation d'accès à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 et AWS des kits SDK.

Utilisation de la console S3

Pour supprimer un octroi d’accès

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Grants.
3. Sur la page Octrois d’accès S3, choisissez la région qui contient l’instance d’octrois d’accès S3 qui vous intéresse.
4. Choisissez Afficher les détails pour cette instance.
5. Sur la page des détails, choisissez l’onglet Octrois.
6. Recherchez l’octroi que vous souhaitez supprimer. Lorsque vous avez localisé l’octroi, choisissez la case d’option correspondante.
7. Sélectionnez Delete (Supprimer). Une boîte de dialogue apparaît pour vous avertir que cette action ne peut pas être annulée. Choisissez à nouveau Supprimer pour supprimer l’octroi.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Supprimer un octroi d’accès

```
aws s3control delete-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

// No response body
```

Utilisation de l'API REST

Pour en savoir plus sur la prise en charge de l’API REST pour la gestion des octrois d’accès, consultez DeleteAccessGrant dans la Référence d’API Amazon Simple Storage Service.

Utilisation des AWS SDK

Cette section fournit des exemples de suppression d'une autorisation d'accès à l'aide des AWS SDK. Pour utiliser l'exemple suivant, remplacez *user input placeholders* par vos propres informations.

Java

Example : Supprimer un octroi d’accès

```
public void deleteAccessGrant() {
DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantResponse deleteResponse =
 s3Control.deleteAccessGrant(deleteRequest);
LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);
}
```

Réponse :

```
DeleteAccessGrantResponse()
```

Demande d’un accès aux données Amazon S3 via les octrois d’accès S3

Après avoir utilisé Amazon S3 Access Grants pour créer une autorisation d'accès qui donne aux principaux AWS Identity and Access Management (IAM), aux identités de votre annuaire d'entreprise ou aux applications autorisées l'accès à vos données S3, vos bénéficiaires peuvent demander des informations d'identification pour accéder à ces données.

Lorsqu'une application ou Service AWS utilise l'opération `GetDataAccess` API pour demander à S3 Access Grants l'accès à vos données S3 au nom d'un bénéficiaire, S3 Access Grants vérifie d'abord que vous avez accordé cet accès d'identité aux données. S3 Access Grants utilise ensuite l'opération AssumeRoleAPI pour obtenir un jeton d'identification temporaire et le transmet au demandeur. Ce jeton d’informations d’identification temporaire est un jeton AWS Security Token Service (AWS STS).

La demande `GetDataAccess` doit inclure le paramètre `target`, qui spécifie la portée des données S3 à laquelle s’appliquent les informations d’identification temporaires. Cette portée `target` peut être identique à la portée de l’octroi ou à un sous-ensemble de cette portée, mais la portée `target` doit être comprise dans la portée de l’octroi accordée au demandeur. La demande doit également spécifier le paramètre `permission` pour indiquer le niveau d’autorisation pour les informations d’identification temporaires, que ce soit `READ`, `WRITE` ou `READWRITE`.

Le demandeur peut spécifier le niveau de privilège du jeton temporaire dans sa demande d’informations d’identification. À l’aide du paramètre `privilege`, le demandeur peut réduire ou augmenter la portée d’accès des informations d’identification temporaires, dans les limites de la portée de l’octroi. La valeur par défaut du paramètre `privilege` est `Default`, ce qui signifie que la portée cible des informations d’identification renvoyées est la portée de l’octroi d’origine. L’autre valeur possible pour `privilege` est `Minimal`. Si la portée `target` est réduite par rapport à la portée de l’octroi d’origine, la portée des informations d’identification temporaires est désactivée pour correspondre à la portée `target`, à condition que la portée `target` soit comprise dans la portée de l’octroi.

Le tableau suivant détaille l’effet du paramètre `privilege` sur deux octrois. Un octroi a la portée `S3://`*`example-s3-bucket1`*`/bob/*`, qui inclut l’intégralité du préfixe `bob/` dans le compartiment *`example-s3-bucket1`*. L’autre octroi a la portée `S3://`*`example-s3-bucket1`*`/bob/reports/*`, qui inclut uniquement le préfixe `bob/reports/` dans le compartiment *`example-s3-bucket1`*.

Portée de l'octroi	Portée demandée	Privilège	Portée renvoyée	Effet
S3://*example-s3-bucket1* /bob/*	*example-s3-bucket1* /bob/*	Default	*example-s3-bucket1* /bob/*	Le demandeur a accès à tous les objets qui ont des noms de clé commençant par le préfixe *bob/* dans le compartiment *example-s3-bucket1* .
S3://*example-s3-bucket1* /bob/*	*example-s3-bucket1* /bob/	Minimal	*example-s3-bucket1* /bob/	Sans le caractère générique * après le nom de préfixe bob/, le demandeur n'a accès qu'à l'objet nommé bob/ dans le compartiment *example-s3-bucket1* . Il n'est pas courant d'avoir un tel objet. Le demandeur n'a accès à aucun autre objet, pas même à ceux ayant des noms de clé commençant par le préfixe bob/.
S3://*example-s3-bucket1* /bob/*	*example-s3-bucket1* /bob/images/*	Minimal	*example-s3-bucket1* /bob/images/*	Le demandeur a accès à tous les objets qui ont des noms de clé commençant par le préfixe *bob/images/** dans le compartiment *example-s3-bucket1* .
S3://*example-s3-bucket1*	*example-s3-bucket1*	Default	*example-s3-bucket1* /bob/reports/*	Le demandeur a accès à tous les objets qui ont des noms de clé commençan

Portée de l'octroi	Portée demandée	Privilège	Portée renvoyée	Effet
`/bob/reports/*`	`/bob/reports/file.txt`			t par le préfixe bob/ `reports` dans le compartiment *`example-s3-bucket1`* , ce qui correspond à la portée de l'octroi correspondant.
`S3://`*`example-s3-bucket1`*`/bob/reports/*`	*`example-s3-bucket1`*`/bob/reports/file.txt`	`Minimal`	*`example-s3-bucket1`*`/bob/reports/file.txt`	Le demandeur a accès uniquement à l'objet doté du nom de clé bob/ `reports/file.txt` dans le compartiment *`example-s3-bucket1`* . Le demandeur n'a accès à aucun autre objet.

Le paramètre `durationSeconds` définit la durée des informations d'identification temporaires, en secondes. La valeur par défaut est de `3600` secondes (1 heure), mais le demandeur (le bénéficiaire) peut spécifier une plage allant de `900` secondes (15 minutes) à `43200` secondes (12 heures). Si le bénéficiaire demande une valeur supérieure à ce maximum, la demande échoue.

> **Note**
>
> Dans votre demande de jeton temporaire, si l'emplacement est un objet, définissez la valeur du paramètre `targetType` dans votre demande sur `Object`. Ce paramètre est obligatoire seulement si l'emplacement est un objet et que le niveau de privilège est `Minimal`. Si l'emplacement est un compartiment ou un préfixe, vous n'avez pas besoin de spécifier ce paramètre.

Pour plus d'informations, consultez GetDataAccessle manuel Amazon Simple Storage Service API Reference.

Vous pouvez demander des informations d'identification temporaires en utilisant AWS Command Line Interface (AWS CLI), l'API REST Amazon S3 et les AWS SDK.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example Demander des informations d'identification temporaires

Requête :

```
aws s3control get-data-access \
--account-id 111122223333 \
--target s3://example-s3-bucket/prefixA* \
--permission READ \
--privilege Default \
--region us-east-2
```

Réponse :

```
{
"Credentials": {
 "AccessKeyId": "Example-key-id",
 "SecretAccessKey": "Example-access-key",
 "SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
    "MatchedGrantTarget": "s3://example-s3-bucket/prefixA**"
}
```

Utilisation de l'API REST

Pour plus d'informations sur la prise en charge par l'API REST d'Amazon S3 pour la demande d'informations d'identification temporaires auprès de S3 Access Grants, consultez GetDataAccessle manuel Amazon Simple Storage Service API Reference.

Utilisation des AWS SDK

Cette section fournit un exemple de la manière dont les bénéficiaires demandent des informations d'identification temporaires à S3 Access Grants à l'aide des AWS SDK.

Java

L’exemple de code suivant renvoie les informations d’identification temporaires que le bénéficiaire utilise pour accéder à vos données S3. Pour utiliser cet exemple de code, remplacez les *user input placeholders* par vos propres informations.

Example Obtenir des informations d’identification temporaires

Requête :

```
public void getDataAccess() {
GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
.accountId("111122223333")
.permission(Permission.READ)
.privilege(Privilege.MINIMAL)
.target("s3://example-s3-bucket/prefixA*")
.build();
GetDataAccessResponse getDataAccessResponse =
 s3Control.getDataAccess(getDataAccessRequest);
LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Réponse :

```
GetDataAccessResponse(
Credentials=Credentials(
AccessKeyId="Example-access-key-id",
SecretAccessKey="Example-secret-access-key",
SessionToken="Example-session-token",
Expiration=2023-06-07T06:55:24Z
))
```

Accédez aux données S3 via un octroi d’accès

Une fois qu’un bénéficiaire a obtenu des informations d’identification temporaires via son octroi d’accès, il peut utiliser ces informations d’identification temporaires pour appeler les opérations d’API Amazon S3 afin d’accéder à vos données.

Les bénéficiaires peuvent accéder aux données S3 en utilisant le AWS Command Line Interface (AWS CLI), les AWS SDK et l'API REST Amazon S3.

En utilisant le AWS CLI

Une fois que le bénéficiaire a obtenu ses informations d'identification temporaires auprès des octrois d'accès S3, il peut configurer un profil avec ces informations d'identification pour récupérer les données.

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Pour utiliser les exemples de commandes suivants, remplacez les *user input placeholders* par vos propres informations.

Example : Configurer un profil

```
aws configure set aws_access_key_id "$accessKey" --profile access-grants-consumer-access-profile
aws configure set aws_secret_access_key "$secretKey" --profile access-grants-consumer-access-profile
aws configure set aws_session_token "$sessionToken" --profile access-grants-consumer-access-profile
```

Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

Example : Obtenir les données S3

Le bénéficiaire peut utiliser la get-object AWS CLI commande pour accéder aux données. Le bénéficiaire peut également utiliser put-objectls, et d'autres AWS CLI commandes S3.

```
aws s3api get-object \
--bucket example-s3-bucket1 \
--key myprefix \
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Utilisation des AWS SDK

Cette section fournit des exemples illustrant comment les bénéficiaires peuvent accéder à vos données S3 à l'aide des kits AWS SDK.

Java

Pour des exemples expliquant comment obtenir des données S3 à l'aide d'informations d'identification temporaires, découvrez comment obtenir un objet à l'aide AWS des SDK et des exemples de code Amazon S3 pour le AWS SDK for Java 2.x.

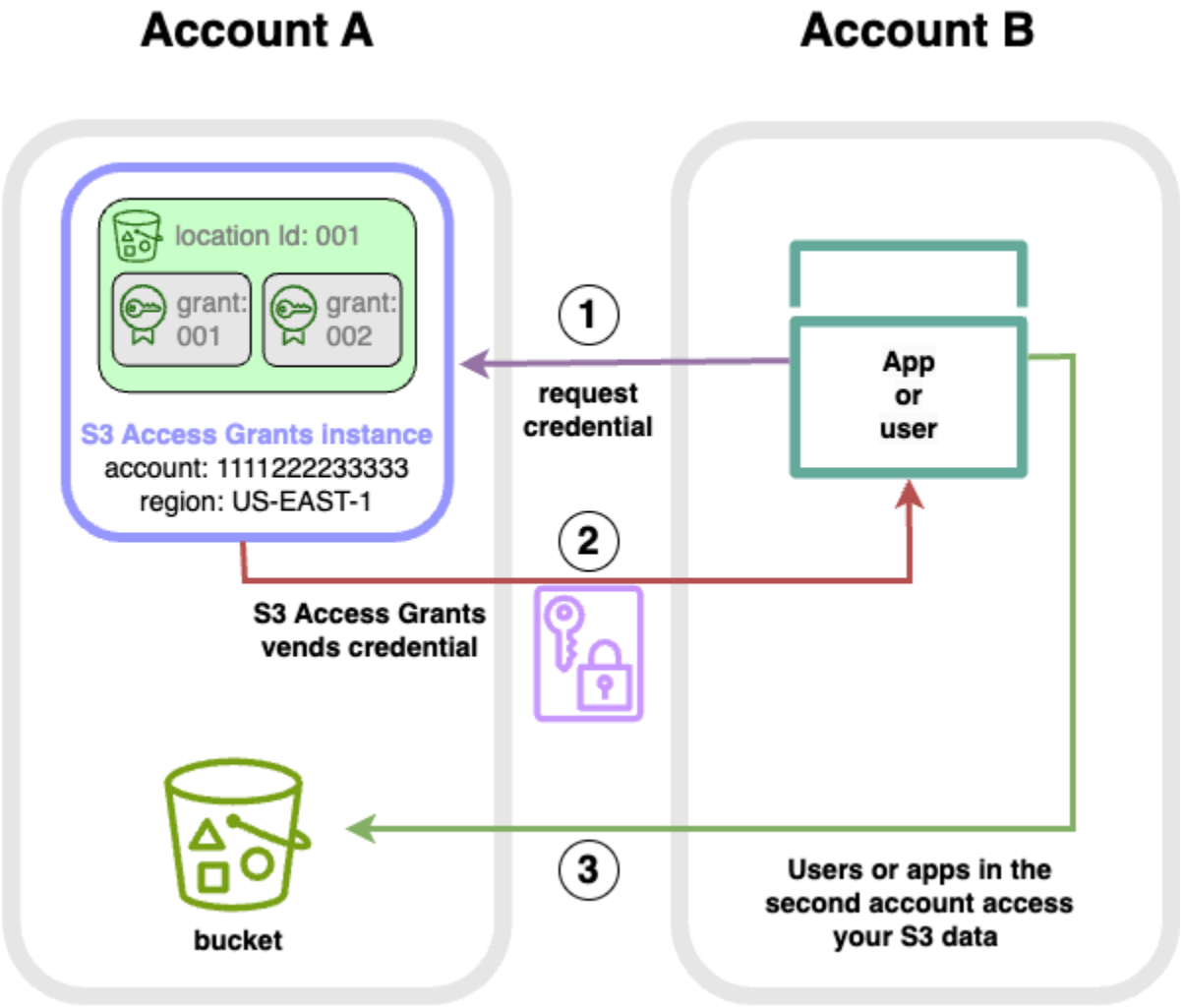
Accès intercompte aux octrois d’accès S3

Avec S3 Access Grants, vous pouvez accorder à Amazon S3 l'accès aux données suivantes :

- AWS Identity and Access Management Identités (IAM) au sein de votre compte
- Identités IAM dans d'autres comptes AWS
- Répertoriez les utilisateurs ou les groupes de votre AWS IAM Identity Center instance

Tout d'abord, configurez l'accès entre comptes pour l'autre compte. Cela inclut l'octroi de l'accès à votre instance S3 Access Grants à l'aide d'une politique de ressources. Accordez ensuite l'accès à vos données S3 (compartiments, préfixes ou objets) en utilisant des autorisations.

Après avoir configuré l'accès entre comptes, l'autre compte peut demander des informations d'accès temporaires à vos données Amazon S3 auprès de S3 Access Grants. L'image suivante montre le flux d'utilisateurs pour l'accès entre comptes S3 via S3 Access Grants :



1. Les utilisateurs ou les applications d'un deuxième compte (B) demandent des informations d'identification à l'instance S3 Access Grants de votre compte (A), où les données Amazon S3 sont stockées. Pour plus d'informations, consultez Demande d'un accès aux données Amazon S3 via les octrois d'accès S3.
2. L'instance S3 Access Grants de votre compte (A) renvoie des informations d'identification temporaires si une autorisation permet au deuxième compte d'accéder à vos données Amazon S3. Pour plus d'informations, consultez the section called "Création d'octrois".
3. Les utilisateurs ou les applications du second compte (B) utilisent les informations d'identification fournies par S3 Access Grants pour accéder aux données S3 de votre compte (A).

Configuration de S3 Access : octroie un accès entre comptes

Pour accorder un accès S3 entre comptes via S3 Access Grants, procédez comme suit :

- Étape 1 : configurez une instance S3 Access Grants dans votre compte, par exemple, l'ID de compte111122223333, où les données S3 sont stockées.
- Étape 2 : Configurez la politique de ressources pour l'instance S3 Access Grants de votre compte 111122223333 afin de donner accès au deuxième compte, par exemple, l'ID du compte444455556666.
- Étape 3 : Configurez les autorisations IAM pour que le principal IAM du deuxième compte demande des informations d'identification 444455556666 à l'instance S3 Access Grants de votre compte. 111122223333
- Étape 4 : Créez une subvention dans votre compte 111122223333 qui donne au principal IAM du deuxième compte l'444455556666accès à certaines des données S3 de votre compte111122223333.

Étape 1 : configurer une instance S3 Access Grants dans votre compte

Tout d'abord, vous devez disposer d'une instance S3 Access Grants dans votre compte 111122223333 pour gérer l'accès à vos données Amazon S3. Vous devez créer une instance S3 Access Grants dans chaque Région AWS endroit où sont stockées les données S3 que vous souhaitez partager. Si vous partagez des données entre plusieurs entités Région AWS, répétez chacune de ces étapes de configuration pour chacune d'entre elles Région AWS. Si vous disposez déjà d'une instance S3 Access Grants dans l' Région AWS endroit où sont stockées vos données S3, passez à l'étape suivante. Si vous n'avez pas configuré d'instance S3 Access Grants, consultez Création d’une instance d’octrois d’accès S3 pour terminer cette étape.

Étape 2 : configurer la politique de ressources pour votre instance S3 Access Grants afin d'accorder un accès entre comptes

Après avoir créé une instance S3 Access Grants dans votre compte 111122223333 pour un accès entre comptes, configurez la politique basée sur les ressources pour l'instance S3 Access Grants de votre compte 111122223333 afin d'accorder un accès entre comptes. L’instance d’octrois d’accès S3 elle-même prend en charge les politiques basées sur les ressources. Lorsque la bonne politique basée sur les ressources est en place, vous pouvez accorder l'accès à des utilisateurs AWS Identity and Access Management (IAM) ou à des rôles provenant d'autres utilisateurs Comptes AWS à votre instance S3 Access Grants. L'accès entre comptes n'accorde que les autorisations (actions) suivantes :

- `s3:GetAccessGrantsInstanceForPrefix`— l'utilisateur, le rôle ou l'application peut récupérer l'instance S3 Access Grants qui contient un préfixe particulier.

- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:GetDataAccess`— l'utilisateur, le rôle ou l'application peut demander des informations d'identification temporaires en fonction de l'accès qui vous a été accordé via S3 Access Grants. Utilisez ces informations d'identification pour accéder aux données S3 auxquelles l'accès vous a été accordé.

Parmi ces autorisations, vous pouvez choisir les autorisations à inclure dans la politique de ressources. Cette politique de ressources sur l'instance S3 Access Grants est une politique normale basée sur les ressources et prend en charge tout ce que le langage de politique IAM prend en charge. Dans la même politique, vous pouvez accorder l'accès à des identités IAM spécifiques dans votre compte`111122223333`, par exemple, en utilisant `aws:PrincipalArn` cette condition, mais vous n'êtes pas obligé de le faire avec S3 Access Grants. Au sein de votre instance S3 Access Grants, vous pouvez plutôt créer des autorisations pour des identités IAM individuelles à partir de votre compte, ainsi que pour l'autre compte. En gérant chaque autorisation d'accès par le biais de S3 Access Grants, vous pouvez augmenter vos autorisations.

Si vous utilisez déjà AWS Resource Access Manager(AWS RAM), vous pouvez l'utiliser pour partager vos `s3:AccessGrants`ressources avec d'autres comptes ou au sein de votre organisation. Consultez la section Utilisation de AWS ressources partagées pour plus d'informations. Si vous ne l'utilisez pas AWS RAM, vous pouvez également ajouter la politique de ressources en utilisant les opérations de l'API S3 Access Grants ou le AWS Command Line Interface (AWS CLI).

Utilisation de la console S3

Nous vous recommandons d'utiliser la console AWS Resource Access Manager (AWS RAM) pour partager vos `s3:AccessGrants` ressources avec d'autres comptes ou au sein de votre organisation. Pour partager des subventions d'accès S3 entre comptes, procédez comme suit :

Pour configurer la politique de ressources de l'instance S3 Access Grants :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Sélectionnez-le dans Région AWS le Région AWS sélecteur.
3. Dans le volet de navigation de gauche, sélectionnez Access Grants.
4. Sur la page de l'instance Access Grants, dans la section Instance dans ce compte, sélectionnez Partager l'instance. Cela vous redirigera vers la AWS RAM console.

5. Sélectionnez Créer un partage de ressources.
6. Suivez les AWS RAM étapes pour créer le partage de ressources. Pour plus d'informations, consultez la section Création d'un partage de ressources dans AWS RAM.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Vous pouvez ajouter la politique de ressources à l'aide de la commande `put-access-grants-instance-resource-policy` CLI.

Si vous souhaitez accorder un accès entre comptes pour l'instance S3 Access Grants se trouvant dans votre compte `111122223333` au second compte`444455556666`, la politique de ressources de l'instance S3 Access Grants de votre compte `111122223333` doit `444455556666` autoriser le second compte à effectuer les actions suivantes :

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dans la politique de ressources de l'instance S3 Access Grants, spécifiez l'ARN de votre instance S3 Access Grants comme étant le`Resource`, et le second compte `444455556666` comme étant le`Principal`. Pour utiliser l'exemple suivant, remplacez les *`espaces réservés saisis par l'utilisateur`* par vos propres informations.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
```

```
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Pour ajouter ou mettre à jour la politique de ressources d'instance S3 Access Grants, utilisez la commande suivante. Lorsque vous utilisez l'exemple de commande suivant, remplacez-le *user input placeholders* par vos propres informations.

Example Ajouter ou mettre à jour la politique de ressources de l'instance S3 Access Grants

```
aws s3control put-access-grants-instance-resource-policy \
--account-id 111122223333 \
--policy file://resourcePolicy.json \
--region us-east-2
{
 "Policy": "{\n
   \"Version\": \"2012-10-17\",\n
   \"Statement\": [{\n
  \"Effect\": \"Allow\",\n
  \"Principal\": {\n
    \"AWS\": \"444455556666\"\n
  },\n
  \"Action\": [\n
    \"s3:ListAccessGrants\",\n
    \"s3:ListAccessGrantsLocations\",\n
    \"s3:GetDataAccess\",\n
    \"s3:GetAccessGrantsInstanceForPrefix\"\n
  ],\n
  \"Resource\": \"arn:aws:s3:us-east-2:111122223333:access-grants/default\"\n
    }\n
   ]\n
   }\n",
 "CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Example Obtenir une politique de ressources d’octrois d’accès S3

Vous pouvez également utiliser la CLI pour obtenir ou supprimer une politique de ressources pour une instance S3 Access Grants.

Pour obtenir une politique de ressources S3 Access Grants, utilisez l'exemple de commande suivant. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

{
"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"arn:aws:iam::111122223333:root\"},\"Action\":[\"s3:ListAccessGrants\",
\"s3:ListAccessGrantsLocations\",\"s3:GetDataAccess\"],\"Resource\":\"arn:aws:s3:us-
east-2:111122223333:access-grants/default\"}]}",
"CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Example Supprimer une politique de ressources d’octrois d’accès S3

Pour supprimer une politique de ressources S3 Access Grants, utilisez l'exemple de commande suivant. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

// No response body
```

Utilisation de l'API REST

Vous pouvez ajouter la politique de ressources à l'aide de l'PutAccessGrantsInstanceResourcePolicy API.

Si vous souhaitez accorder un accès entre comptes pour l'instance S3 Access Grants se trouvant dans votre compte 111122223333 au second compte444455556666, la politique de ressources de l'instance S3 Access Grants de votre compte 111122223333 doit 444455556666 autoriser le second compte à effectuer les actions suivantes :

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`

- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dans la politique de ressources de l'instance S3 Access Grants, spécifiez l'ARN de votre instance S3 Access Grants comme étant le`Resource`, et le second compte `444455556666` comme étant le`Principal`. Pour utiliser l'exemple suivant, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Vous pouvez ensuite utiliser l'PutAccessGrantsInstanceResourcePolicy API pour configurer la politique.

Pour plus d'informations sur la prise en charge de l'API REST pour mettre à jour, obtenir ou supprimer une politique de ressources pour une instance S3 Access Grants, consultez les sections suivantes du manuel Amazon Simple Storage Service API Reference :

- PutAccessGrantsInstanceResourcePolicy
- GetAccessGrantsInstanceResourcePolicy
- DeleteAccessGrantsInstanceResourcePolicy

Utilisation des AWS SDK

Cette section fournit des exemples du AWS SDK expliquant comment configurer votre politique de ressources S3 Access Grants afin d'accorder à un deuxième AWS compte l'accès à certaines de vos données S3.

Java

Ajoutez, mettez à jour, obtenez ou supprimez une politique de ressources pour gérer l’accès intercompte à votre instance d’octrois d’accès S3.

Example Ajouter ou mettre à jour une politique de ressources d'instance S3 Access Grants

Si vous souhaitez accorder un accès entre comptes pour l'instance S3 Access Grants se trouvant dans votre compte `111122223333` au second compte`444455556666`, la politique de ressources de l'instance S3 Access Grants de votre compte `111122223333` doit `444455556666` autoriser le second compte à effectuer les actions suivantes :

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Dans la politique de ressources de l'instance S3 Access Grants, spécifiez l'ARN de votre instance S3 Access Grants comme étant le`Resource`, et le second compte `444455556666` comme étant le`Principal`. Pour utiliser l'exemple suivant, remplacez les *espaces réservés saisis par l'utilisateur* par vos propres informations.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
```

```
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Pour ajouter ou mettre à jour une politique de ressources d'instance S3 Access Grants, utilisez l'exemple de code suivant :

```
public void putAccessGrantsInstanceResourcePolicy() {
 PutAccessGrantsInstanceResourcePolicyRequest putRequest =
 PutAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .policy(RESOURCE_POLICY)
 .build();
 PutAccessGrantsInstanceResourcePolicyResponse putResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
 LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
 }
```

Réponse :

```
PutAccessGrantsInstanceResourcePolicyResponse(
 Policy={
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
 },
 "Action": [
 "s3:ListAccessGrants",
 "s3:ListAccessGrantsLocations",
 "s3:GetDataAccess",
 "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
 }]
 }
 )
```

Example Obtenir une politique de ressources d'octrois d'accès S3

Pour obtenir une politique de ressources S3 Access Grants, utilisez l'exemple de code suivant. Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

```
public void getAccessGrantsInstanceResourcePolicy() {
 GetAccessGrantsInstanceResourcePolicyRequest getRequest =
 GetAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 GetAccessGrantsInstanceResourcePolicyResponse getResponse =
 s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
 LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
 }
```

Réponse :

```
GetAccessGrantsInstanceResourcePolicyResponse(
 Policy={"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::444455556666:root"},"Action":
["s3:ListAccessGrants","s3:ListAccessGrantsLocations","s3:GetDataAccess"],"Resource":"arn:aw
east-2:111122223333:access-grants/default"}]},
 CreatedAt=2023-06-15T22:54:44.319Z
 )
```

Example Supprimer une politique de ressources d'octrois d'accès S3

Pour supprimer une politique de ressources S3 Access Grants, utilisez l'exemple de code suivant. Pour utiliser l'exemple de commande suivant, remplacez *user input placeholders* par vos propres informations.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
 DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
 DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
 LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
 }
```

Réponse :

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Étape 3 : Accorder aux identités IAM d'un deuxième compte l'autorisation d'appeler l'instance S3 Access Grants de votre compte

Une fois que le propriétaire des données Amazon S3 a configuré la politique inter-comptes pour l'instance S3 Access Grants en compte111122223333, le propriétaire du second compte 444455556666 doit créer une politique basée sur l'identité pour ses utilisateurs ou rôles IAM, et le propriétaire doit leur donner accès à l'instance S3 Access Grants. Dans la politique basée sur l'identité, incluez une ou plusieurs des actions suivantes, en fonction de ce qui est accordé dans la politique de ressources d'instance S3 Access Grants et des autorisations que vous souhaitez accorder :

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Selon le AWS modèle d'accès entre comptes, les utilisateurs ou rôles IAM du second compte 444455556666 doivent explicitement disposer d'une ou de plusieurs de ces autorisations. Par exemple, accordez l'`s3:GetDataAccess`autorisation afin que l'utilisateur ou le rôle IAM puisse appeler l'instance S3 Access Grants associée au compte 111122223333 pour demander des informations d'identification.

Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
  "Action": [
   "s3:GetDataAccess",
  ],
   "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
```

```
  }
 ]
}
```

Pour plus d'informations sur la modification de la stratégie basée sur l'identité IAM, consultez la section Modification des politiques IAM dans le guide.AWS Identity and Access Management

Étape 4 : Créez une autorisation dans l'instance S3 Access Grants de votre compte qui donne à l'identité IAM du second compte l'accès à certaines de vos données S3

Pour la dernière étape de configuration, vous pouvez créer une autorisation dans l'instance S3 Access Grants de votre compte 111122223333 qui donne accès à l'identité IAM du deuxième compte 444455556666 à certaines données S3 de votre compte. Vous pouvez le faire à l'aide de la console Amazon S3, de la CLI, de l'API et des kits de développement logiciel (SDK). Pour plus d'informations, consultez Création d'octrois.

Dans l'autorisation, spécifiez l' AWS ARN de l'identité IAM du deuxième compte et spécifiez à quel emplacement dans vos données S3 (un compartiment, un préfixe ou un objet) vous accordez l'accès. Cet emplacement doit déjà être enregistré auprès de votre instance S3 Access Grants. Pour plus d'informations, consultez Enregistrement d'un emplacement. Vous pouvez éventuellement spécifier un sous-préfixe. Par exemple, si l'emplacement auquel vous accordez l'accès est un compartiment et que vous souhaitez limiter davantage l'accès à un objet spécifique de ce compartiment, transmettez le nom de la clé de l'objet dans le `S3SubPrefix` champ. Ou si vous souhaitez limiter l'accès aux objets du compartiment dont les noms de clé commencent par un préfixe spécifique`2024-03-research-results/`, tel que « pass `S3SubPrefix=2024-03-research-results/` ».

Voici un exemple de commande CLI permettant de créer une autorisation d'accès pour une identité dans le second compte. Pour plus d'informations, consultez Création d'octrois. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix=prefixA* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::444455556666:role/data-consumer-1
```

Après avoir configuré l'accès entre comptes, l'utilisateur ou le rôle du second compte peut effectuer les opérations suivantes :

- Appels `ListAccessGrantsInstances` pour répertorier les instances S3 Access Grants partagées avec ce service via AWS RAM. Pour plus d'informations, consultez Affichage des détails d'une instance d'octrois d'accès S3.
- Demande des informations d'identification temporaires à S3 Access Grants. Pour plus d'informations sur la procédure à suivre pour effectuer ces demandes, consultezDemande d'un accès aux données Amazon S3 via les octrois d'accès S3.

Utilisation de AWS balises avec S3 Access Grants

Dans les octrois d'accès Amazon S3, les balises présentent des caractéristiques similaires à celles des balises d'objet dans Amazon S3. Chaque balise est une paire clés-valeurs. Les ressources que vous pouvez baliser dans les octrois d'accès S3 sont les instances, les emplacements et les octrois.

> **Note**
>
> Le balisage dans les octrois d'accès S3 utilise des opérations d'API différentes de celles du balisage d'objets. Les octrois d'accès S3 utilisent les opérations d'API TagResource, UntagResource et ListTagsForResource, où une ressource peut être une instance d'octrois d'accès S3, un emplacement enregistré ou un octroi d'accès.

Comme avec les balises d'objet, les limitations suivantes s'appliquent :

- Vous pouvez ajouter des balises à de nouvelles ressources d'octrois d'accès S3 lorsque vous les créez, ou vous pouvez ajouter des balises à des ressources existantes.
- Vous pouvez associer jusqu'à 10 balises à une ressource. Si plusieurs balises sont associées à la même ressource, elles doivent avoir des clés de balise uniques.
- Une clé d'étiquette peut comporter jusqu'à 128 caractères Unicode et les valeurs d'étiquette peuvent comporter jusqu'à 256 caractères Unicode. Les balises sont représentées en interne au format UTF-16. Dans le format UTF-16, les caractères occupent une ou deux positions de caractère.
- Les clés et les valeurs sont sensibles à la casse.

Pour plus d'informations sur les restrictions liées aux balises, consultez Restrictions encadrant les balises définies par l'utilisateur dans le Guide de l'utilisateur AWS Billing .

Vous pouvez baliser les ressources dans S3 Access Grants à l'aide de AWS Command Line Interface (AWS CLI), de l'API REST Amazon S3 ou AWS des SDK.

En utilisant le AWS CLI

Pour l'installer AWS CLI, reportez-vous à la section Installation du AWS CLI dans le guide de AWS Command Line Interface l'utilisateur.

Vous pouvez baliser une ressource d'octrois d'accès S3 lorsque vous la créez ou après l'avoir créée. Les exemples suivants montrent comment baliser une instance d'octrois d'accès S3 ou en annuler le balisage. Vous pouvez effectuer des opérations similaires pour des emplacements enregistrés et des octrois d'accès.

Pour utiliser les exemples de commandes suivants, remplacez les *user input placeholders* par vos propres informations.

Example : Créer une instance d'octrois d'accès S3 avec des balises

```
aws s3control create-access-grants-instance \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region us-east-2 \
 --tags Key=tagKey1,Value=tagValue1
```

Réponse :

```
 {
    "CreatedAt": "2023-10-25T01:09:46.719000+00:00",
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Example : Baliser une instance d'octrois d'accès S3 déjà créée

```
aws s3control tag-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2 \
```

```
--tags Key=tagKey2,Value=tagValue2
```

Example : Répertorier les balises pour l'instance d'octrois d'accès S3

```
aws s3control list-tags-for-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2
```

Réponse :

```
{
    "Tags": [
        {
            "Key": "tagKey1",
            "Value": "tagValue1"
        },
        {
            "Key": "tagKey2",
            "Value": "tagValue2"
        }
    ]
}
```

Example : Annuler le balisage de l'instance d'octrois d'accès S3

```
aws s3control untag-resource \
 --account-id 111122223333 \
 --resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
 --profile access-grants-profile \
 --region us-east-2 \
 --tag-keys "tagKey2"
```

Utilisation de l'API REST

Vous pouvez utiliser l'API Amazon S3 pour définir des balises, annuler le balisage ou répertorier les balises pour une instance d'octrois d'accès S3, un emplacement enregistré ou un octroi d'accès. Pour en savoir plus sur la prise en charge de l'API REST pour la gestion des balises d'octrois d'accès S3, consultez les sections suivantes dans la Référence d'API Amazon Simple Storage Service :

- TagResource
- UntagResource
- ListTagsForResource

Limitations des octrois d’accès S3

Les octrois d’accès S3 présentent les limitations suivantes :

> **Note**
>
> Si votre cas d'utilisation dépasse ces limites, contactez le AWS support pour demander des limites plus élevées.

Instance d’octrois d’accès S3

Vous pouvez créer une instance S3 Access Grants Région AWS par compte. Consultez Création d’une instance d’octrois d’accès S3.

Emplacement d’octrois d’accès S3

Vous pouvez enregistrer 1 000 emplacements d’octrois d’accès S3 par instance d’octrois d’accès S3. Consultez Enregistrement d’un emplacement d’octrois d’accès S3.

Octroi

Vous pouvez créer 100 000 octrois par instance d’octrois d’accès S3. Consultez Création d’un octroi.

Intégrations des octrois d’accès S3

Les subventions d'accès S3 peuvent être utilisées avec les AWS services et fonctionnalités suivants. Cette page sera mise à jour au fur et à mesure que de nouvelles intégrations seront disponibles.

AWS IAM Identity Center

Propagation d’identité approuvée entre applications

Amazon EMR

Lancement d’un cluster Amazon EMR avec les octrois d’accès S3

Amazon EMR on EKS

- Lancement d’un cluster Amazon EMR sur EKS avec les octrois d’accès S3

Application Amazon EMR sans serveur

- Lancement d’une application Amazon EMR sans serveur avec les octrois d’accès S3

Amazon Athena

- Utilisation des groupes de travail Athena compatibles avec IAM Identity Center

Gestion des accès à l'aide des listes ACL

Les listes de contrôle d'accès (ACL) sont l'une des options basées sur les ressources que vous pouvez utiliser pour gérer l'accès à vos buckets et à vos objets. Vous pouvez utiliser les ACL pour accorder des autorisations de lecture et d'écriture de base à d'autres personnes. Comptes AWS La gestion des autorisations grâce aux listes ACL a ses limites.

Par exemple, vous ne pouvez accorder des autorisations qu'à d'autres personnes Comptes AWS ; vous ne pouvez pas accorder d'autorisations aux utilisateurs de votre compte. Vous ne pouvez pas accorder d'autorisations conditionnelles ou refuser explicitement des autorisations. Les listes ACL conviennent aux scénarios spécifiques. Par exemple, si le propriétaire d'un compartiment autorise d'autres personnes Comptes AWS à télécharger des objets, les autorisations relatives à ces objets ne peuvent être gérées Compte AWS que par le propriétaire de l'objet à l'aide de l'ACL de l'objet.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d’informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Pour plus d'informations sur les listes ACL, consultez les rubriques suivantes :

Rubriques

- Présentation de la liste de contrôle d'accès (ACL)
- Configuration des listes ACL
- Exemples de politiques pour les ACL

Présentation de la liste de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) d'Amazon S3 vous permettent d'accéder aux compartiments et aux objets. Chaque compartiment et objet possède une liste ACL qui lui est attachée comme sous-ressource. Il définit le Comptes AWS ou les groupes auxquels l'accès est accordé et le type d'accès. Lors de la réception d'une demande sur une ressource, Amazon S3 vérifie la liste ACL correspondante pour s'assurer que le demandeur possède les autorisations d'accès nécessaires.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès

à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Lorsque vous créez un compartiment ou un objet, Amazon S3 crée une liste ACL par défaut qui octroie au propriétaire de la ressource le contrôle total sur celle-ci. Ce comportement est illustré dans l'exemple de liste ACL de compartiment suivant (la liste ACL d'objet par défaut possède la même structure) :

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

L'exemple de liste ACL inclut un élément `Owner` qui identifie le propriétaire par l'ID d'utilisateur canonique du Compte AWS. Pour obtenir des instructions pour trouver votre ID d'utilisateur canonique, consultez Trouver un nom d'utilisateur Compte AWS canonique. L'élément `Grant` identifie le bénéficiaire (soit un groupe prédéfini, Compte AWS soit un groupe prédéfini) et l'autorisation accordée. Cette liste ACL par défaut possède un élément `Grant` pour le propriétaire. Vous accordez des autorisations en ajoutant des éléments `Grant`, avec chaque accord identifiant le bénéficiaire et l'autorisation.

> **Note**
>
> Une liste ACL peut avoir 100 accords maximum.

Rubriques

Qui est un bénéficiaire ?

Un bénéficiaire peut être un Compte AWS ou l'un des groupes Amazon S3 prédéfinis. Vous autorisez l' Compte AWS utilisation de l'adresse e-mail ou de l'ID utilisateur canonique. Toutefois, si vous fournissez une adresse e-mail dans la demande d'accord, Amazon S3 trouve l'ID d'utilisateur canonique pour ce compte et l'ajoute à la liste ACL. Les ACL qui en résultent contiennent toujours l'ID utilisateur canonique du Compte AWS, et non l'adresse e-mail du. Compte AWS

Lorsque vous accordez des droits d'accès, vous spécifiez chaque bénéficiaire sous la forme d'une paire *`type`*`="`*`value`*`"`, où le *`type`* est l'un des suivants :

- `id`— Si la valeur spécifiée est l'ID utilisateur canonique d'un Compte AWS
- `uri` : si vous accordez des autorisations à un groupe prédéfini
- `emailAddress` : si la valeur spécifiée est l'adresse e-mail d'un Compte AWS

Important

L'utilisation d'adresses e-mail pour spécifier un bénéficiaire est prise en charge uniquement dans les Régions AWS suivantes :

- US East (N. Virginia)
- USA Ouest (Californie du Nord)
- US West (Oregon)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asia Pacific (Tokyo)
- Europe (Ireland)
- South America (São Paulo)

Pour obtenir la liste de toutes les régions et de tous les points de terminaison Amazon S3 pris en charge, consultez Régions et points de terminaison dans Référence générale d'Amazon Web Services.

Example Exemple : adresse e-mail

Par exemple, l'`x-amz-grant-read`en-tête suivant autorise les adresses e-mail Comptes AWS identifiées par les adresses e-mail à lire les données des objets et leurs métadonnées :

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

Warning

Lorsque vous accordez à d'autres personnes l' Comptes AWS accès à vos ressources, sachez qu'elles Comptes AWS peuvent déléguer leurs autorisations aux utilisateurs de leurs comptes. Il s'agit d'un accès entre comptes. Pour en savoir plus sur l'utilisation de l'accès intercompte, consultez Création d'un rôle pour la délégation d'autorisations à un utilisateur IAM dans le Guide de l'utilisateur IAM.

Trouver un nom d'utilisateur Compte AWS canonique

L'ID d'utilisateur canonique est associé au Compte AWS. Cet ID est composé d'une longue chaîne de caractères, par exemple :

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Pour plus d'informations sur la façon de trouver l'ID utilisateur canonique de votre compte, voir Trouver l'ID utilisateur canonique correspondant à votre compte Compte AWS dans le Guide de référence de gestion de AWS compte.

Vous pouvez également rechercher l'ID utilisateur canonique d'un Compte AWS en lisant l'ACL d'un bucket ou d'un objet auquel il Compte AWS possède des autorisations d'accès. Lorsqu'une personne Compte AWS obtient des autorisations par le biais d'une demande de subvention, une entrée de subvention est ajoutée à l'ACL avec l'ID utilisateur canonique du compte.

> **Note**
>
> Si vous rendez votre compartiment public (non recommandé), n'importe quel utilisateur non authentifié pourra y charger des objets. Ces utilisateurs anonymes ne disposent pas d'un Compte AWS. Lorsqu'un utilisateur anonyme charge un objet dans votre compartiment, Amazon S3 ajoute un ID utilisateur canonique spécial (`65a011a29cdf8ec533ec3d1ccaae921c`) comme propriétaire de l'objet dans la liste ACL. Pour plus d'informations, consultez Propriété du compartiment et de l'objet Amazon S3.

Groupes prédéfinis Amazon S3

Amazon S3 possède un ensemble de groupes prédéfinis. Lorsque l'accès à un compte est accordé à un groupe, vous spécifiez l'un des URI Amazon S3 au lieu de l'ID d'utilisateur canonique. Amazon S3 fournit les groupes prédéfinis suivants :

- Groupe des utilisateurs authentifiés – Représenté par `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

  Ce groupe représente tout le monde Comptes AWS. L'autorisation d'accès à ce groupe permet Compte AWS à n'importe qui d'accéder à la ressource. Toutefois, toutes les demandes doivent être signées (authentifiées).

> **Warning**
>
> Lorsque vous accordez l'accès au groupe Utilisateurs authentifiés, n'importe quel utilisateur AWS authentifié dans le monde peut accéder à votre ressource.

- Groupe de tous les utilisateurs – Représenté par `http://acs.amazonaws.com/groups/global/AllUsers`.

  Une autorisation d'accès à ce groupe permet à tout le monde d'accéder à la ressource. Les demandes peuvent être signées (authentifiées) ou non (anonymes). Les demandes non signées omettent l'en-tête Authentification dans la demande.

  > **Warning**
  >
  > Nous vous recommandons vivement de ne pas jamais accorder au groupe de tous les utilisateurs les autorisations `WRITE`, `WRITE_ACP` ou `FULL_CONTROL`. Par exemple, bien que les autorisations `WRITE` ne permettent pas aux non-propriétaires de remplacer ou de supprimer des objets existants, les autorisations `WRITE` permettent toujours à quiconque de stocker des objets dans votre compartiment, ce pour quoi vous êtes facturé. Pour plus de détails sur ces autorisations, consultez la section suivante Quelles autorisations puis-je octroyer ?.

- Groupe de livraison des journaux – Représenté par `http://acs.amazonaws.com/groups/s3/LogDelivery`.

  Une autorisation `WRITE` sur un compartiment permet à ce groupe d'écrire des journaux d'accès au serveur (consultez Enregistrement de demandes avec journalisation des accès au serveur) dans le compartiment.

> **Note**
>
> Lorsque vous utilisez des ACL, le bénéficiaire peut être un Compte AWS ou l'un des groupes Amazon S3 prédéfinis. Toutefois, le bénéficiaire ne peut pas être un utilisateur IAM. Pour plus d'informations sur les utilisateurs et autorisations AWS dans IAM, consultez Utilisation d'AWS Identity and Access Management.

Quelles autorisations puis-je octroyer ?

Le tableau suivant répertorie l'ensemble des autorisations qu'Amazon S3 prend en charge dans une liste ACL. Toutes les autorisations de liste ACL sont identiques pour les listes ACL d'objet et de compartiment. Toutefois, selon le contexte (liste ACL de compartiment ou liste ACL d'objet), ces autorisations de liste ACL accordent des autorisations pour des opérations spécifiques de compartiment ou d'objet. Le tableau répertorie les autorisations et décrit ce qu'elles signifient pour des objets et des compartiments.

Pour plus d’informations sur les autorisations ACL dans la console Amazon S3, consultez Configuration des listes ACL.

Autorisations ACL

Autorisation	Lorsqu'elles sont accordées sur un compartiment	Lorsqu'elles sont accordées sur un objet
`READ`	Elles permettent au bénéficiaire de lister les objets dans le compartiment	Elles permettent au bénéficiaire de lire les données de l'objet et ses métadonnées
`WRITE`	Elles permettent au bénéficiaire de créer des objets dans le compartiment. Pour les propriétaires de compartiments et d'objets existants, elles permettent également de supprimer et de remplacer ces objets.	Ne s’applique pas
`READ_ACP`	Elles permettent au bénéficiaire de lire la liste ACL du compartiment	Elles permettent au bénéficiaire de lire la liste ACL de l'objet
`WRITE_ACP`	Elles permettent au bénéficiaire d'écrire la liste ACL pour le compartiment applicable	Elles permettent au bénéficiaire d'écrire la liste ACL pour l'objet applicable
`FULL_CONTROL`	Elle accorde au bénéficiaire les autorisations `READ`, `WRITE`, `READ_ACP` et `WRITE_ACP` sur le compartiment	Elle accorde au bénéficiaire les autorisations `READ`, `READ_ACP` et `WRITE_ACP` sur l'objet

> **Warning**
>
> Soyez vigilant lorsque vous accordez des autorisations d'accès à vos compartiments et objets S3. Par exemple, l'octroi de l'accès `WRITE` à un compartiment permet au bénéficiaire de créer des objets dans le compartiment. Nous vous recommandons vivement de lire l'ensemble de la section Présentation de la liste de contrôle d'accès (ACL) avant d'accorder des autorisations.

Mappage des autorisations de liste ACL et de stratégie d'accès

Comme illustré dans la table précédente, une liste ACL permet uniquement d'accorder un ensemble limité d'autorisations, comparé au nombre d'autorisations configurables dans une stratégie d'accès (consultez Actions politiques pour Amazon S3). Chacune de ces autorisations permet une ou plusieurs opérations Amazon S3.

Le tableau suivant montre comment chacune des autorisations de liste ACL est mappée aux autorisations de stratégie d'accès correspondantes. Comme vous pouvez le voir, une stratégie d'accès accorde plus d'autorisations qu'une liste ACL. Vous utilisez une liste ACL principalement pour accorder des autorisations de lecture/écriture de base, similaires aux autorisations de système de fichiers. Pour plus d'informations sur le moment où utiliser une ACL, consultez Identity and Access Management pour Amazon S3.

Pour plus d'informations sur les autorisations ACL dans la console Amazon S3, consultez Configuration des listes ACL.

Autorisation de liste ACL	Autorisations de stratégie d'accès correspondantes lorsqu'une autorisation de liste ACL est accordée sur un compartiment	Autorisations de stratégie d'accès correspondantes lorsqu'une autorisation de liste ACL est accordée sur un objet
`READ`	`s3:ListBucket` , `s3:ListBucketVersions` et `s3:ListBucketMultipartUploads`	`s3:GetObject` et `s3:GetObjectVersion`
`WRITE`	`s3:PutObject` Le propriétaire du compartiment peut créer, remplacer et supprimer	Ne s'applique pas

Autorisation de liste ACL	Autorisations de stratégie d'accès correspondantes lorsqu'une autorisation de liste ACL est accordée sur un compartiment	Autorisations de stratégie d'accès correspondantes lorsqu'une autorisation de liste ACL est accordée sur un objet
	n'importe quel objet dans le compartiment, et le propriétaire de l'objet bénéficie du `FULL_CONTROL` sur son objet. De plus, lorsque le bénéficiaire est le propriétaire du compartiment, l'accord d'une autorisation `WRITE` dans la liste ACL d'un compartiment permet d'exécuter l'action `s3:DeleteObjectVersion` sur n'importe quelle version de ce compartiment.	
`READ_ACP`	`s3:GetBucketAcl`	`s3:GetObjectAcl` et `s3:GetObjectVersionAcl`
`WRITE_ACP`	`s3:PutBucketAcl`	`s3:PutObjectAcl` et `s3:PutObjectVersionAcl`
`FULL_CONTROL`	Équivaut à accorder les autorisations de liste ACL `READ`, `WRITE`, `READ_ACP` et `WRITE_ACP` . Par conséquent, cette autorisation de liste ACL est mappée à une combinaison d'autorisations de stratégie d'accès correspondantes.	Équivaut à accorder les autorisations de liste ACL `READ`, `READ_ACP` et `WRITE_ACP` . Par conséquent, cette autorisation de liste ACL est mappée à une combinaison d'autorisations de stratégie d'accès correspondantes.

Clés de condition

Lorsque vous accordez des autorisations de stratégie d'accès, vous pouvez utiliser des clés de condition pour limiter la valeur de l'ACL sur un objet à l'aide d'une stratégie de compartiment. Les clés de contexte suivantes correspondent aux listes ACL. Vous pouvez utiliser ces clés de contexte pour exiger l'utilisation d'une liste ACL spécifique dans une demande :

- `s3:x-amz-grant-read` - Exiger un accès en lecture.
- `s3:x-amz-grant-write` - Exiger un accès en écriture.
- `s3:x-amz-grant-read-acp` - Exiger un accès en lecture à l'ACL du compartiment.
- `s3:x-amz-grant-write-acp` - Exiger un accès en écriture à l'ACL du compartiment.
- `s3:x-amz-grant-full-control` - Exiger un contrôle total.
- `s3:x-amz-acl` - Exiger une Liste ACL prête à l'emploi.

Pour des exemples de stratégies impliquant des en-têtes spécifiques à une liste ACL, consultez Octroi de s3 : PutObject autorisation assortie d'une condition obligeant le propriétaire du bucket à obtenir le contrôle total. Pour obtenir la liste complète des clés de condition spécifiques à Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Valeurs **`aclRequired`** pour les demandes Amazon S3 courantes

Pour identifier les demandes Amazon S3 qui ont nécessité des listes ACL pour l'autorisation, vous pouvez utiliser la valeur `aclRequired` dans les journaux d'accès du serveur Amazon S3 ou AWS CloudTrail. La `aclRequired` valeur qui apparaît dans les journaux CloudTrail d'accès au serveur Amazon S3 dépend des opérations appelées et de certaines informations concernant le demandeur, le propriétaire de l'objet et le propriétaire du compartiment. Si aucune ACL n'est requise, ou si vous définissez l'ACL `bucket-owner-full-control` prédéfinie, ou si les demandes sont autorisées par votre politique de compartiment, la chaîne de `aclRequired` valeur est « `-` » dans les journaux d'accès au serveur Amazon S3 et est absente dans CloudTrail.

Les tableaux suivants répertorient les `aclRequired` valeurs attendues dans les journaux CloudTrail d'accès au serveur Amazon S3 pour les différentes opérations d'API Amazon S3. Vous pouvez utiliser ces informations pour comprendre quelles opérations Amazon S3 dépendent des listes ACL pour l'autorisation. Dans les tables suivantes, A, B et C représentent les différents comptes associés au demandeur, au propriétaire de l'objet et au propriétaire du compartiment. Les entrées suivies d'un astérisque (*) indiquent l'un des comptes A, B ou C.

> **Note**
>
> Les opérations `PutObject` de la table suivante, sauf indication contraire, indiquent les demandes qui ne définissent pas de liste ACL, sauf si la liste ACL est `bucket-owner-`

`full-control`. Une valeur nulle pour `aclRequired` indique qu'elle `aclRequired` est absente des AWS CloudTrail journaux.

`aclRequired`valeurs pour CloudTrail

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`GetObject`	A	A	A	Oui ou Non	null	Accès dans le même compte
	A	B	A	Oui ou Non	null	Accès au même compte avec le propriétaire du compartiment obligatoire
	A	A	B	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	A	B	Non	Oui	Accès intercomp

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
						te reposant sur la liste ACL
	A	A	B	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	B	B	Non	Oui	Accès intercompte reposant sur la liste ACL
	A	B	C	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	B	C	Non	Oui	Accès intercompte reposant sur la liste ACL

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`PutObject`	A	Ne s'applique pas	A	Oui ou Non	null	Accès dans le même compte
	A	Ne s'applique pas	B	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
`PutObject` avec une liste ACL (sauf pour `bucket-owner-full-control`)	*	Ne s'applique pas	*	Oui ou Non	Oui	Demande autorise la liste ACL

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`ListObjects`	A	Ne s'applique pas	A	Oui ou Non	null	Accès dans le même compte
	A	Ne s'applique pas	B	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
`DeleteObject`	A	Ne s'applique pas	A	Oui ou Non	null	Accès dans le même compte

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
	A	Ne s'applique pas	B	Oui	null	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
`PutObjectAcl`	*	*	*	Oui ou Non	Oui	Demande autorise la liste ACL
`PutBucketAcl`	*	Ne s'applique pas	*	Oui ou Non	Oui	Demande autorise la liste ACL

> **Note**
>
> Les opérations `REST.PUT.OBJECT` de la table suivante, sauf indication contraire, indiquent les demandes qui ne définissent pas de liste ACL, sauf si la liste ACL est `bucket-owner-full-control`. Une chaîne de valeur `aclRequired` « `-` » indique une valeur nulle dans les journaux d'accès au serveur Amazon S3.

Valeurs **`aclRequired`** pour les journaux d'accès au serveur Amazon S3

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`REST.GET.OBJECT`	A	A	A	Oui ou Non	-	Accès dans le même compte
	A	B	A	Oui ou Non	-	Accès au même compte avec le propriétaire du compartiment obligatoire
	A	A	B	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	A	B	Non	Oui	Accès intercompte reposant sur la liste ACL

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
	A	B	B	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	B	B	Non	Oui	Accès intercompte reposant sur la liste ACL
	A	B	C	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	B	C	Non	Oui	Accès intercompte reposant sur la liste ACL

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`REST.PUT.OBJECT`	A	Ne s'applique pas	A	Oui ou Non	-	Accès dans le même compte
	A	Ne s'applique pas	B	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
`REST.PUT.OBJECT` avec une liste ACL (sauf pour `bucket-owner-full-control`)	*	Ne s'applique pas	*	Oui ou Non	Oui	Demande autorise la liste ACL

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **`aclRequired`**	Raison
`REST.GET.BUCKET`	A	Ne s'applique pas	A	Oui ou Non	-	Accès dans le même compte
	A	Ne s'applique pas	B	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
`REST.DELETE.OBJECT`	A	Ne s'applique pas	A	Oui ou Non	-	Accès dans le même compte

Nom de l'opération	Demandeur	Propriétaire de l'objet	Propriétaire du compartiment	La politique du compartiment autorise l'accès	Valeur **aclRequired**	Raison
	A	Ne s'applique pas	B	Oui	-	Accès intercompte accordé par la politique du compartiment
	A	Ne s'applique pas	B	Non	Oui	Accès intercompte reposant sur la liste ACL
REST.PUT.ACL	*	*	*	Oui ou Non	Oui	Demande autorise la liste ACL

Exemple de liste ACL

L'exemple de liste ACL suivant sur un compartiment identifie le propriétaire de la ressource et un ensemble d'accords. Le format est la représentation XML d'une liste ACL dans l'API REST Amazon S3. Le propriétaire du compartiment possède l'accès FULL_CONTROL sur la ressource. En outre, l'ACL montre comment les autorisations sont accordées sur une ressource à deux Comptes AWS, identifiés par un ID utilisateur canonique, et à deux des groupes Amazon S3 prédéfinis décrits dans la section précédente.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
```

```
 <Owner>
   <ID>Owner-canonical-user-ID</ID>
   <DisplayName>display-name</DisplayName>
 </Owner>
 <AccessControlList>
   <Grant>
     <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
       <ID>Owner-canonical-user-ID</ID>
       <DisplayName>display-name</DisplayName>
     </Grantee>
     <Permission>FULL_CONTROL</Permission>
   </Grant>

   <Grant>
     <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
       <ID>user1-canonical-user-ID</ID>
       <DisplayName>display-name</DisplayName>
     </Grantee>
     <Permission>WRITE</Permission>
   </Grant>

   <Grant>
     <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="CanonicalUser">
       <ID>user2-canonical-user-ID</ID>
       <DisplayName>display-name</DisplayName>
     </Grantee>
     <Permission>READ</Permission>
   </Grant>

   <Grant>
     <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
       <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
     </Grantee>
     <Permission>READ</Permission>
   </Grant>
   <Grant>
     <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
       <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
     </Grantee>
     <Permission>WRITE</Permission>
   </Grant>
```

```
  </AccessControlList>
</AccessControlPolicy>
```

Liste ACL prête à l'emploi

Amazon S3 prend en charge un ensemble d'accords prédéfinis, appelées listes ACL prédéfinies. Chaque liste ACL prête à l'emploi possède un ensemble prédéfini de bénéficiaires et d'autorisations. Le tableau suivant liste l'ensemble des listes ACL prêtes à l'emploi et les accords prédéfinis associés.

Liste ACL prête à l'emploi	S'applique à	Autorisations ajoutées à la liste ACL
`private`	Compartiment et objet	Le propriétaire obtient l'accès `FULL_CONTROL`. Personne d'autre ne possède les droits d'accès (par défaut).
`public-read`	Compartiment et objet	Le propriétaire obtient l'accès `FULL_CONTROL`. Le groupe `AllUsers` (consultez Qui est un bénéficiaire ?) obtient l'accès `READ`.
`public-read-write`	Compartiment et objet	Le propriétaire obtient l'accès `FULL_CONTROL`. Le groupe `AllUsers` obtient l'accès `READ` et `WRITE`. L'accord de ce type d'accès sur un compartiment n'est généralement pas recommandé.
`aws-exec-read`	Compartiment et objet	Le propriétaire obtient l'accès `FULL_CONTROL`. Amazon EC2 obtient l'accès `READ` pour faire une demande `GET` sur une solution groupée Amazon Machine Image (AMI) issu d'Amazon S3.
`authenticated-read`	Compartiment et objet	Le propriétaire obtient l'accès `FULL_CONTROL`. Le groupe `AuthenticatedUsers` obtient l'accès `READ`.
`bucket-owner-read`	Objet	Le propriétaire de l'objet obtient l'accès `FULL_CONTROL`. Le propriétaire du compartiment obtient l'accès `READ`. Si vous spécifiez cette ACL prédéfinie lors de la création du compartiment, Amazon S3 l'ignore.

Liste ACL prête à l'emploi	S'applique à	Autorisations ajoutées à la liste ACL
`bucket-owner-full-control`	Objet	Le propriétaire de l'objet et celui du compartiment obtiennent l'accès `FULL_CONTROL` sur l'objet. Si vous spécifiez cette ACL prédéfinie lors de la création du compartiment, Amazon S3 l'ignore.
`log-delivery-write`	Compartiment	Le groupe `LogDelivery` obtient les autorisations `WRITE` et `READ_ACP` sur le compartiment. Pour plus d'informations sur les journaux, consultez (Enregistrement de demandes avec journalisation des accès au serveur).

> **Note**
>
> Vous pouvez uniquement spécifier l'une des listes ACL prêtes à l'emploi dans la demande.

Vous pouvez spécifier une liste ACL prédéfinie dans votre demande grâce à l'en-tête `x-amz-acl`. Lorsqu'Amazon S3 reçoit une demande contenant une liste ACL prédéfinie, il ajoute les accords prédéfinis à la liste ACL de la ressource.

Configuration des listes ACL

Cette section explique comment gérer les autorisations d'accès des compartiments et objets S3 à l'aide des listes de contrôle d'accès (ACL). Vous pouvez ajouter des autorisations à votre ACL de ressources à l' AWS Management Console aide de l'API REST AWS Command Line Interface (CLI) ou AWS des SDK.

Les autorisations de compartiment et d'objet sont indépendantes les unes des autres. un objet n'hérite pas des autorisations de son compartiment. Par exemple, si vous créez un compartiment et accordez un accès en écriture à un utilisateur, vous ne pouvez pas accéder à ses objets sauf s'il vous accorde explicitement l'accès.

Vous pouvez accorder des autorisations à d'autres Compte AWS utilisateurs ou à des groupes prédéfinis. L'utilisateur ou le groupe auquel vous accordez des autorisations est le « bénéficiaire ».

Par défaut, le propriétaire, qui a créé le compartiment, dispose des autorisations complètes. Compte AWS

Chaque autorisation accordée pour un utilisateur ou un groupe ajoute une entrée dans la liste ACL associée au compartiment. Les listes ACL répertorient le bénéficiaire et l'autorisation accordée.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

> **Warning**
>
> Nous vous recommandons vivement d'éviter d'accorder un accès en écriture aux groupes Tout le monde (accès public) ou Utilisateurs authentifiés (tous les utilisateurs AWS authentifiés). Pour en savoir plus sur les effets de l'octroi d'un accès en écriture à ces groupes, veuillez consulter Groupes prédéfinis Amazon S3.

Utilisation de la console S3 pour définir des autorisations ACL pour un compartiment

La console affiche les autorisations d'accès combinées pour les bénéficiaires en double. Pour consulter la liste complète des ACL, utilisez l'API REST Amazon S3 ou AWS CLI les AWS SDK.

Le tableau suivant présente les autorisations ACL que vous pouvez configurer pour les compartiments dans la console Amazon S3.

Autorisations ACL de la console Amazon S3 pour les compartiments

Autorisation de la console	Autorisation de liste ACL	Accès
Objets – Liste	`READ`	Elles permettent au bénéficiaire de répertorier les objets dans le compartiment
Objets – Écriture	`WRITE`	Elles permettent au bénéficiaire de créer des objets dans le compartiment. Pour les propriétaires de compartiments et d'objets existants, elles permettent également de supprimer et de remplacer ces objets.
ACL de compartiment – Lecture	`READ_ACP`	Elles permettent au bénéficiaire de lire la liste ACL du compartiment.
ACL de compartiment – Écriture	`WRITE_ACP`	Elles permettent au bénéficiaire d'écrire la liste ACL pour le compartiment applicable.
Tout le monde (accès public) : Objets – Liste	`READ`	Elles accordent un accès public en lecture pour les objets se trouvant dans le compartiment. Lorsque vous accordez l'accès à la liste à Tout le monde (accès public), quiconque dans le monde peut accéder aux objets présents dans le compartiment.
Tout le monde (accès public) : ACL	`READ_ACP`	Elles accordent un accès public en lecture pour l'ACL de compartiment. Lorsque vous accordez l'accès en lecture à Tout le monde (accès public), quiconque dans le monde peut accéder à l'ACL de compartiment.

Autorisation de la console	Autorisation de liste ACL	Accès
de compartiment – Lecture		

Pour plus d'informations sur les autorisations ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Pour définir des autorisations de listes ACL pour un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez définir des autorisations.
3. Choisissez Permissions.
4. Sous Liste de contrôle d'accès, choisissez Modifier.

   Vous pouvez modifier les autorisations ACL suivantes pour le compartiment :

   Objets

   - List – Permet au bénéficiaire de lister les objets dans le compartiment.
   - Write (Écriture) – Permet au bénéficiaire de créer des objets dans le compartiment. Pour les propriétaires de compartiments et d'objets existants, elles permettent également de supprimer et de remplacer ces objets.

Dans la console S3, vous pouvez uniquement accorder un accès en écriture au groupe de mise à disposition de journaux S3 et au propriétaire du compartiment (le vôtre Compte AWS). Nous vous recommandons vivement de ne pas accorder l'accès en écriture aux autres bénéficiaires. Toutefois, si vous devez accorder un accès en écriture, vous pouvez utiliser les AWS CLI AWS SDK ou l'API REST.

ACL du compartiment

- Read – Permet au bénéficiaire de lire la liste ACL du compartiment.
- Write – Permet au bénéficiaire d'écrire la liste ACL pour le compartiment applicable.

5. Pour modifier les autorisations du propriétaire du bucket, à côté du propriétaire du bucket (votre Compte AWS), effacez ou sélectionnez l'une des autorisations ACL suivantes :

   - Objets — Liste ou Écriture
   - ACL de compartiment — Lecture ou Écriture

   Le propriétaire fait référence à l'utilisateur Utilisateur racine d'un compte AWS, et non à un utilisateur AWS Identity and Access Management IAM. Pour plus d'informations sur l'utilisateur root, consultez la section Utilisateur racine d'un compte AWS du Guide de l'utilisateur IAM.

6. Pour octroyer ou annuler des autorisations pour le grand public (tout le monde sur Internet), en regard de Tout le monde (accès public), désactivez ou sélectionnez l'une des autorisations ACL suivantes :

   - Objets — Liste
   - ACL de compartiment — Lecture

   > **Warning**
   >
   > Soyez vigilant lorsque vous accordez au groupe Tout le monde l'accès public à votre compartiment S3. Lorsque vous accordez l'accès à ce groupe, tout le monde peut accéder à votre compartiment. Nous vous recommandons vivement de ne jamais accorder un type d'accès en écriture public quel qu'il soit à votre compartiment S3.

7. Pour accorder ou annuler des autorisations à toute personne disposant d'un Compte AWSgroupe d'utilisateurs authentifiés (toute personne possédant un Compte AWS), effacez ou sélectionnez l'une des autorisations ACL suivantes :

   - Objets — Liste
   - ACL de compartiment — Lecture

8. Pour octroyer ou annuler des autorisations à Amazon S3 pour écrire des journaux d'accès au serveur dans le compartiment, sous Groupe de mise à disposition des journaux S3, désactivez ou sélectionnez l'une des autorisations ACL suivantes :

   - Objets — Liste ou Écriture
   - ACL de compartiment — Lecture ou Écriture

   Si un compartiment est configuré en tant que compartiment cible (les journaux d'accès y seront stockés), les autorisations sur ce compartiment doivent autoriser le groupe Livraison des journaux à disposer d'un accès en écriture sur le compartiment. Lorsque vous activez la journalisation des accès serveur sur un compartiment, la console Amazon S3 accorde au groupe Log Delivery (Livraison des journaux) un droit d'accès en écriture sur le compartiment que vous avez choisi pour la réception des journaux. Pour en savoir plus sur la journalisation des accès au serveur, consultez Activation de la journalisation des accès au serveur Amazon S3.

9. Pour accorder l'accès à une autre Compte AWS personne, procédez comme suit :

   a. Choisissez Ajouter un bénéficiaire.

   b. Dans la zone Bénéficiaire, saisissez l'ID canonique de l'autre Compte AWS.

   c. Sélectionnez l'une des autorisations ACL suivantes :

      - Objets — Liste ou Écriture
      - ACL de compartiment — Lecture ou Écriture

> **Warning**
>
> Lorsque vous accordez à d'autres personnes l' Comptes AWS accès à vos ressources, sachez qu'elles Comptes AWS peuvent déléguer leurs autorisations aux utilisateurs de leurs comptes. Il s'agit d'un accès entre comptes. Pour en savoir plus sur l'utilisation de

l'accès intercompte, consultez Création d'un rôle pour la délégation d'autorisations à un utilisateur IAM dans le Guide de l'utilisateur IAM.

10. Pour supprimer l'accès à un autre Compte AWS, sous Accès pour les autres Comptes AWS, choisissez Supprimer.
11. Choisissez Enregistrer pour enregistrer les modifications.

Utilisation de la console S3 pour définir des autorisations ACL pour un objet

La console affiche les autorisations d'accès combinées pour les bénéficiaires en double. Pour consulter la liste complète des ACL, utilisez l'API REST Amazon S3 ou AWS CLI les AWS SDK. Le tableau suivant présente les autorisations ACL que vous pouvez configurer pour les objets dans la console Amazon S3.

Autorisations ACL de la console Amazon S3 pour les objets

Autorisation de la console	Autorisation de liste ACL	Accès
Objet – Lecture	`READ`	Elles permettent au bénéficiaire de lire les données de l'objet et ses métadonnées.
ACL de l'objet – Lecture	`READ_ACP`	Elles permettent au bénéficiaire de lire la liste ACL de l'objet.
ACL de l'objet – Écriture	`WRITE_ACP`	Elles permettent au bénéficiaire d'écrire la liste ACL pour l'objet applicable

Pour plus d'informations sur les autorisations ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

Important

Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur

`AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Pour définir des autorisations de liste ACL pour un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, sélectionnez le nom de l'objet pour lequel vous souhaitez définir des autorisations.
4. Choisissez Permissions.
5. Sous Liste de contrôle d'accès (ACL), sélectionnez Modifier.

   Vous pouvez modifier les autorisations ACL suivantes pour l'objet :

   Objet

   - Read – Permet au bénéficiaire de lire les données de l'objet et ses métadonnées.

   Liste ACL de l'objet

   - Read – Permet au bénéficiaire de lire la liste ACL de l'objet.
   - Write – Permet au bénéficiaire d'écrire la liste ACL pour l'objet applicable. Dans la console S3, vous ne pouvez accorder l'accès en écriture qu'au propriétaire du compartiment (le vôtre Compte AWS). Nous vous recommandons vivement de ne pas accorder l'accès en écriture aux autres bénéficiaires. Toutefois, si vous devez accorder un accès en écriture, vous pouvez utiliser les AWS CLI AWS SDK ou l'API REST.
6. Vous pouvez gérer les autorisations d'accès aux objets pour :

   a. Accès pour le propriétaire de l'objet

      Le propriétaire fait référence à Utilisateur racine d'un compte AWS, et non à un utilisateur AWS Identity and Access Management IAM. Pour plus d'informations sur l'utilisateur root, consultez la section Utilisateur racine d'un compte AWS du Guide de l'utilisateur IAM.

      Pour modifier les autorisations d'accès aux objets du propriétaire, sous Accès pour le propriétaire de l'objet, sélectionnez Votre AWS compte (propriétaire).

Activez les cases à cocher des autorisations que vous souhaitez modifier, puis choisissez Enregistrer.

b. Accès pour les autres Comptes AWS

Pour accorder des autorisations à un autre AWS utilisateur Compte AWS, sous Accès pour les autres Comptes AWS, sélectionnez Ajouter un compte. Dans le champ Entrez un identifiant, entrez l'identifiant canonique de l' AWS utilisateur auquel vous souhaitez accorder des autorisations d'objets. Pour plus d'informations sur la recherche d'un identifiant canonique, consultez la section Vos Compte AWS identifiants dans le. Référence générale d'Amazon Web Services Vous pouvez ajouter jusqu'à 99 utilisateurs.

Activez les cases à cocher des autorisations que vous souhaitez accorder à l'utilisateur, puis choisissez Enregistrer. Pour afficher des informations sur les autorisations, choisissez les icônes d'aide.

c. Accès public

Pour permettre au grand public (tout le monde) d'accéder à votre objet, sous Accès public, sélectionnez Tout le monde. Si vous accordez des autorisations d'accès public, tout le monde peut accéder à l'objet.

Activez les cases à cocher des autorisations que vous souhaitez accorder, puis choisissez Enregistrer.

> ⚠ Warning
>
> - Soyez vigilant lorsque vous accordez au groupe Everyone (Tout le monde) l'accès anonyme à vos objets Amazon S3. Lorsque vous accordez l'accès à ce groupe, tout le monde peut accéder à votre objet. Si vous avez besoin d'accorder l'accès à tout le monde, nous vous recommandons vivement d'octroyer uniquement des autorisations Lecture d'objet.
> - Nous vous recommandons de ne pas accorder des autorisations d'écriture sur l'objet au groupe Tout le monde. Si vous le faites, n'importe qui peut remplacer les autorisations de liste ACL pour l'objet.

Utilisation des AWS SDK

Cette section fournit des exemples de configuration des attributions de liste ACL sur les compartiments et les objets.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Java

Cette section fournit des exemples de configuration des attributions de liste ACL sur les compartiments et les objets. Le premier exemple crée un compartiment avec une liste ACL prête à l'emploi (voir Liste ACL prête à l'emploi), crée une liste personnalisée d'attributions d'autorisation, puis remplace l'ACL prête à l'emploi avec une ACL contenant les attributions personnalisées. Le second exemple montre comment modifier une ACL à l'aide de la méthode `AccessControlList.grantPermission()`.

Example Créer un compartiment et spécifier une liste ACL conservée qui octroie une autorisation au groupe de mise à disposition du journal S3

Cet exemple crée un compartiment. Dans le demande, l'exemple spécifie une liste ACL prête à l'emploi qui attribue au groupe Log Delivery l'autorisation d'écrire des journaux sur le compartiment.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
```

```
import java.util.ArrayList;

public class CreateBucketWithACL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String userEmailForReadPermission = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Create a bucket with a canned ACL. This ACL will be replaced by the
            // setBucketAcl()
            // calls below. It is included here for demonstration purposes.
            CreateBucketRequest createBucketRequest = new
 CreateBucketRequest(bucketName, clientRegion.getName())
                    .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
            s3Client.createBucket(createBucketRequest);

            // Create a collection of grants to add to the bucket.
            ArrayList<Grant> grantCollection = new ArrayList<Grant>();

            // Grant the account owner full control.
            Grant grant1 = new Grant(new
 CanonicalGrantee(s3Client.getS3AccountOwner().getId()),
                    Permission.FullControl);
            grantCollection.add(grant1);

            // Grant the LogDelivery group permission to write to the bucket.
            Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
            grantCollection.add(grant2);

            // Save grants by replacing all current ACL grants with the two we just
 created.
            AccessControlList bucketAcl = new AccessControlList();
            bucketAcl.grantAllPermissions(grantCollection.toArray(new Grant[0]));
            s3Client.setBucketAcl(bucketName, bucketAcl);

            // Retrieve the bucket's ACL, add another grant, and then save the new
 ACL.
            AccessControlList newBucketAcl = s3Client.getBucketAcl(bucketName);
```

```
            Grant grant3 = new Grant(new
 EmailAddressGrantee(userEmailForReadPermission), Permission.Read);
            newBucketAcl.grantAllPermissions(grant3);
            s3Client.setBucketAcl(bucketName, newBucketAcl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Mettre à jour la liste ACL sur un objet existant

L'exemple met à jour l'ACL sur un objet. L'exemple exécute les tâches suivantes :

- Extrait l'ACL d'un objet
- Efface la liste ACL en supprimant toutes les autorisations existantes
- Ajoute deux autorisations : plein accès au propriétaire, et WRITE_ACP (voir Quelles autorisations puis-je octroyer ?) à un utilisateur identifié par une adresse e-mail
- Enregistre l'ACL sur l'objet

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.EmailAddressGrantee;
import com.amazonaws.services.s3.model.Permission;

import java.io.IOException;
```

```
public class ModifyACLExistingObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String emailGrantee = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Get the existing object ACL that we want to modify.
            AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);

            // Clear the existing list of grants.
            acl.getGrantsAsList().clear();

            // Grant a sample set of permissions, using the existing ACL owner for
 Full
            // Control permissions.
            acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),
 Permission.FullControl);
            acl.grantPermission(new EmailAddressGrantee(emailGrantee),
 Permission.WriteAcp);

            // Save the modified ACL back to the object.
            s3Client.setObjectAcl(bucketName, keyName, acl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Example Créer un compartiment et spécifier une liste ACL conservée qui octroie une autorisation au groupe de mise à disposition du journal S3

Cet exemple C# crée un compartiment. Dans le demande, le code spécifie aussi une liste ACL prête à l'emploi qui attribue au groupe Log Delivery l'autorisation d'écrire les journaux sur le compartiment.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingBucketACLTest
    {
        private const string newBucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateBucketUseCannedACLAsync().Wait();
        }

        private static async Task CreateBucketUseCannedACLAsync()
        {
            try
            {
                // Add bucket (specify canned ACL).
                PutBucketRequest putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
```

```
                    BucketRegion = S3Region.EUW1, // S3Region.US,
                                                  // Add canned ACL.
                    CannedACL = S3CannedACL.LogDeliveryWrite
                };
                PutBucketResponse putBucketResponse = await
 client.PutBucketAsync(putBucketRequest);

                // Retrieve bucket ACL.
                GetACLResponse getACLResponse = await client.GetACLAsync(new
 GetACLRequest
                {
                    BucketName = newBucketName
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("S3 error occurred. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Example Mettre à jour la liste ACL sur un objet existant

L'exemple C# met à jour l'ACL sur un objet existant. L'exemple exécute les tâches suivantes :

- Extrait l'ACL d'un objet.
- Efface la liste ACL en supprimant toutes les autorisations existantes.
- Ajoute deux autorisations : plein accès au propriétaire, et WRITE_ACP à un utilisateur identifié par une adresse e-mail.
- Enregistre l'ACL en envoyant une demande `PutAcl`.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key name ***";
        private const string emailAddress = "*** email address ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            TestObjectACLTestAsync().Wait();
        }
        private static async Task TestObjectACLTestAsync()
        {
            try
            {
                    // Retrieve the ACL for the object.
                    GetACLResponse aclResponse = await client.GetACLAsync(new
 GetACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName
                    });

                    S3AccessControlList acl = aclResponse.AccessControlList;

                    // Retrieve the owner (we use this to re-add permissions after
 we clear the ACL).
                    Owner owner = acl.Owner;

                    // Clear existing grants.
                    acl.Grants.Clear();
```

```
                    // Add a grant to reset the owner's full permission (the
 previous clear statement removed all permissions).
                    S3Grant fullControlGrant = new S3Grant
                    {
                        Grantee = new S3Grantee { CanonicalUser = owner.Id },
                        Permission = S3Permission.FULL_CONTROL

                    };

                    // Describe the grant for the permission using an email address.
                    S3Grant grantUsingEmail = new S3Grant
                    {
                        Grantee = new S3Grantee { EmailAddress = emailAddress },
                        Permission = S3Permission.WRITE_ACP
                    };
                    acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
 grantUsingEmail });

                    // Set a new ACL.
                    PutACLResponse response = await client.PutACLAsync(new
 PutACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        AccessControlList = acl
                    });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Utilisation de l'API REST

L'API d'Amazon S3 vous permet de configurer une liste ACL lorsque vous créez un compartiment ou un objet. Amazon S3 fournit une API pour configurer une liste ACL sur un compartiment ou un objet existant. Ces API vous fournissent les méthodes suivantes pour configurer une liste ACL :

- Configurer la liste ACL grâce aux en-têtes de demande – Lorsque vous envoyez une demande pour créer une ressource (compartiment ou objet), vous configurez une liste ACL grâce aux en-têtes de demande. Grâce à ces en-têtes, vous pouvez spécifier une liste ACL prête à l'emploi ou des accords (en identifiant explicitement le bénéficiaire et les autorisations).
- Configurer la liste ACL grâce au corps de la demande – Lorsque vous envoyez une demande pour configurer une liste ACL sur une ressource existante, vous pouvez configurer la liste ACL dans l'en-tête ou le corps de la demande.

Pour plus d'informations sur la prise en charge de l'API REST pour la gestion des listes ACL, consultez les sections suivantes dans la Référence d'API Amazon Simple Storage Service :

- GET Bucket acl
- PUT Bucket acl
- GET Object acl
- PUT Object acl
- PUT Object
- PUT Bucket
- PUT Object - Copy
- Lancement du chargement partitionné

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

En-têtes de demande spécifiques à une liste de contrôle d'accès (ACL)

Vous pouvez utiliser des en-têtes pour accorder des autorisations basées sur la liste de contrôle d'accès (ACL). Par défaut, tous les objets sont privés. Seul le propriétaire dispose d'un contrôle d'accès complet. Lorsque vous ajoutez un nouvel objet, vous pouvez accorder des autorisations à des individus Comptes AWS ou à des groupes prédéfinis définis par Amazon S3. Ces autorisations sont ensuite ajoutées à la liste de contrôle d'accès (ACL) sur l'objet. Pour plus d'informations, consultez Présentation de la liste de contrôle d'accès (ACL).

Avec cette opération, vous pouvez accorder des autorisations d'accès en utilisant l'une des deux méthodes suivantes :

- Liste ACL prête à l'emploi (**`x-amz-acl`**) : Amazon S3 prend en charge un ensemble de listes ACL prédéfinies, appelées « listes ACL prêtes à l'emploi ». Chaque liste ACL prête à l'emploi possède un ensemble prédéfini de bénéficiaires et d'autorisations. Pour plus d'informations, consultez Liste ACL prête à l'emploi.
- Autorisations d'accès — Pour accorder explicitement des autorisations d'accès à des groupes Comptes AWS ou à des groupes spécifiques, utilisez les en-têtes suivants. Chaque en-tête correspond à des autorisations spécifiques prises en charge par Amazon S3 dans une liste ACL. Pour plus d'informations, consultez Présentation de la liste de contrôle d'accès (ACL). Dans l'en-tête, vous spécifiez une liste de bénéficiaires qui obtiennent l'autorisation spécifique.
  - x-amz-grant-read
  - x-amz-grant-write
  - x-amz-grant-read-acp
  - x-amz-grant-write-acp
  - x-amz-grant-full-contrôle

En utilisant le AWS CLI

Pour plus d'informations sur la gestion des ACL à l'aide de AWS CLI, consultez put-bucket-acllla référence des AWS CLI commandes.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du

compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Exemples de politiques pour les ACL

Vous pouvez utiliser des clés de condition dans les politiques de compartiment pour contrôler l'accès à Amazon S3.

Rubriques

- Octroi de s3 : PutObject autorisation assortie d'une condition obligeant le propriétaire du bucket à obtenir le contrôle total
- Octroi de s3 : PutObject autorisation avec une condition sur l' x-amz-acl en-tête

Octroi de s3 : PutObject autorisation assortie d'une condition obligeant le propriétaire du bucket à obtenir le contrôle total

L'opération PUT Object autorise les en-têtes spécifiques à la liste de contrôle d'accès (ACL) que vous pouvez utiliser pour accorder des autorisations basées sur les listes ACL. En utilisant ces clés, le propriétaire du compartiment peut définir une condition pour nécessiter des autorisations d'accès spécifiques quand l'utilisateur charge un objet.

Supposons que le Compte A possède un compartiment et que l'administrateur du compte souhaite accorder à Dave, un utilisateur du Compte B, des autorisations pour charger des objets. Par défaut, les objets que Dave charge appartiennent au Compte B et le Compte A n'a aucune autorisation sur ces objets. Le propriétaire du compartiment payant les factures, il souhaite toutes les autorisations sur les objets que Dave charge. L'administrateur du Compte A peut accomplir cela en octroyant l'autorisation `s3:PutObject` à Dave, avec une condition que la demande inclue des en-têtes spécifiques à la liste ACL, qui soit accorde explicitement une autorisation complète, soit utilise une liste ACL prédéfinie. Pour plus d'informations, consultez Objet PUT.

Exiger l' x-amz-full-control en-tête

Vous pouvez exiger l'en-tête `x-amz-full-control` dans la demande avec une autorisation de contrôle total pour le propriétaire du compartiment. La stratégie de compartiment suivante octroie

l'autorisation `s3:PutObject` à l'utilisateur Dave avec une condition utilisant la clé de condition `s3:x-amz-grant-full-control`, qui nécessite que la demande inclut l'en-tête `x-amz-full-control`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
        }
      }
    }
  ]
}
```

> **Note**
>
> Cet exemple porte sur l'autorisation entre comptes. Toutefois, si Dave (qui obtient l'autorisation) appartient au propriétaire du Compte AWS bucket, cette autorisation conditionnelle n'est pas nécessaire. En effet, le compte parent auquel Dave appartient possède des objets que l'utilisateur charge.

Ajouter un refus explicite

La stratégie de compartiment précédente octroie l'autorisation conditionnelle à l'utilisateur Dave du compte B. Tandis que cette stratégie est appliquée, il est possible pour Dave d'obtenir la même autorisation sans aucune condition par le biais d'autres stratégies. Par exemple, Dave peut appartenir à un groupe et vous octroyez l'autorisation `s3:PutObject` de groupe sans aucune condition. Pour éviter de telles failles d'autorisation, vous pouvez écrire une stratégie d'accès plus stricte en ajoutant un refus explicite. Dans cet exemple, vous refusez explicitement à l'utilisateur Dave l'autorisation de

chargement s'il n'inclut pas les en-têtes nécessaires dans la demande octroyant des autorisations complète au propriétaire du compartiment. Le refus explicite a toujours priorité sur n'importe quelle autre autorisation accordée. Vous trouverez, ci-après, l'exemple révisé de stratégie d'accès avec un refus explicite ajouté.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        }
    ]
}
```

Testez la politique à l'aide du AWS CLI

Si vous en avez deux Comptes AWS, vous pouvez tester la politique à l'aide du AWS Command Line Interface (AWS CLI). Vous joignez la politique et utilisez les informations d'identification de

Dave pour tester l'autorisation à l'aide de la AWS CLI `put-object` commande suivante. Vous fournissez les informations d'identification de Dave en ajoutant le paramètre `--profile`. Vous octroyez l'autorisation de contrôle complet au propriétaire du compartiment en ajoutant le paramètre `--grant-full-control`. Pour plus d'informations sur la configuration et l'utilisation du AWS CLI, consultezDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Exiger l' x-amz-acl en-tête

Vous pouvez exiger l'en-tête `x-amz-acl` avec une liste ACL prédéfinie octroyant une autorisation de contrôle complet au propriétaire du compartiment. Pour demander l'en-tête `x-amz-acl` dans la demande, vous pouvez remplacer la paire de clé-valeur dans le bloc `Condition` et spécifier la clé de condition `s3:x-amz-acl` comme indiqué dans l'exemple suivant.

```
"Condition": {
    "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
    }
}
```

Pour tester l'autorisation à l'aide du AWS CLI, vous devez spécifier le `--acl` paramètre. Il ajoute AWS CLI ensuite l'`x-amz-acl`en-tête lorsqu'il envoie la demande.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --acl "bucket-owner-full-control" --profile AccountBadmin
```

Octroi de s3 : PutObject autorisation avec une condition sur l' x-amz-acl en-tête

La politique de compartiment suivante accorde l'`s3:PutObject`autorisation à deux personnes Comptes AWS si la demande inclut l'`x-amz-acl`en-tête rendant l'objet lisible par le public. Le bloc `Condition` utilise la condition `StringEquals` et elle dispose d'une paire de clé-valeur, `"s3:x-amz-acl":["public-read"]`, pour évaluation. Dans la paire de clé-valeur, `s3:x-amz-acl` est une clé propre à Amazon S3, comme indiqué par le préfixe `s3:`.

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
         "Sid":"AddCannedAcl",
            "Effect":"Allow",
         "Principal": {
                "AWS": [
                "arn:aws:iam::Account1-ID:root",
    "arn:aws:iam::Account2-ID:root"
    ]
            },
         "Action":"s3:PutObject",
            "Resource": ["arn:aws:s3:::awsexamplebucket1/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl":["public-read"]
                }
            }
        }
    ]
}
```

> **Important**
>
> Les conditions ne sont pas toutes logiques pour toutes les actions. Par exemple, il est logique d'inclure une condition `s3:LocationConstraint` sur une stratégie qui octroie l'autorisation Amazon S3 `s3:CreateBucket`. Cependant, il n'est pas logique d'inclure cette condition dans une politique qui accorde l'autorisation `s3:GetObject`. Amazon S3 peut rechercher les erreurs sémantiques pour ce type qui impliquent des conditions spécifiques à Amazon S3. Cependant, si vous créez une stratégie pour un rôle ou un utilisateur IAM et que vous incluez une condition Amazon S3 non valide sémantiquement, aucune erreur n'est rapportée car IAM ne peut pas valider les conditions Amazon S3.

Blocage de l'accès public à votre stockage Amazon S3

La fonction du blocage de l'accès public Amazon S3 fournit des paramètres pour les points d'accès, les compartiments et les comptes afin de vous aider à gérer l'accès public aux ressources Amazon S3. Par défaut, les nouveaux compartiments, points d'accès et objets n'autorisent pas l'accès public. Toutefois, les utilisateurs peuvent modifier les stratégies de compartiment, les stratégies de point d'accès ou les autorisations d'objet pour autoriser l'accès public. Les paramètres de la fonctionnalité

de blocage de l'accès public S3 remplacent ces stratégies et autorisations pour que vous puissiez restreindre l'accès public à ces ressources.

Grâce au blocage de l'accès public Amazon S3, les administrateurs de compte et les propriétaires de compartiment configurent facilement les contrôles centralisés, afin de restreindre l'accès public à leurs ressources Amazon S3. Ces contrôles sont appliqués quel que soit le mode de création de ces ressources.

Pour obtenir des instructions sur la configuration du blocage de l'accès public, consultez Configuration du blocage d'accès public.

Quand Amazon S3 reçoit une demande d'accès à un compartiment ou à un objet, il détermine si le paramètre de blocage de l'accès public est défini pour le compartiment ou le compte du propriétaire de compartiment. Si la demande a été effectuée via un point d'accès, Amazon S3 vérifie également les paramètres de blocage de l'accès public pour le point d'accès. S'il existe un paramètre de blocage de l'accès public interdisant l'accès demandé, Amazon S3 rejette la demande.

Le blocage de l'accès public Amazon S3 fournit quatre paramètres. Ces paramètres sont indépendants et peuvent être fournis sous n'importe quelle combinaison. Chaque paramètre peut être appliqué à un point d'accès, à un compartiment ou à un Compte AWS entier. Si les paramètres de blocage de l'accès public pour le point d'accès, le compartiment ou le compte diffèrent, Amazon S3 applique la combinaison la plus restrictive des paramètres du point d'accès, du compartiment et du compte.

Quand Amazon S3 évalue si une opération est interdite par un paramètre de blocage de l'accès public, il rejette toute demande qui irait à l'encontre d'un paramètre de point d'accès, de compartiment ou de compte.

> **Important**
>
> Un accès public est accordé aux compartiments et objets via les listes de contrôle d'accès (ACL), les stratégies de compartiment ou les deux. Pour être sûr que l'accès public à tous vos points d'accès, compartiments et objets Amazon S3 est bloqué, nous vous recommandons d'activer les quatre paramètres liés au blocage de l'accès public pour votre compte. Ces paramètres bloquent l'accès public pour tous les compartiments et points d'accès présents et futurs.
>
> Avant d'appliquer ces paramètres, vérifiez que vos applications fonctionnent correctement sans accès public. Si vous avez besoin d'un certain niveau d'accès public à vos compartiments ou objets, par exemple pour héberger un site web statique comme décrit dans

Hébergement d'un site Web statique à l'aide d'Amazon S3,vous pouvez personnaliser les paramètres individuels afin de les adapter à vos cas d'utilisation du stockage. L'activation du blocage de l'accès public permet de protéger vos ressources en empêchant l'accès public d'être accordé par le biais des politiques de ressources ou des listes de contrôle d'accès (ACL) directement associées aux ressources S3. Outre l'activation du blocage de l'accès public, examinez attentivement les politiques suivantes pour vous assurer qu'elles n'accordent pas d'accès public :

- Politiques basées sur l'identité associées aux AWS principaux associés (par exemple, les rôles IAM)
- Politiques basées sur les AWS ressources associées (par exemple, clés AWS Key Management Service (KMS))

Note

- Vous pouvez activer les paramètres de blocage de l'accès public uniquement pour les points d'accès, les compartiments et les Comptes AWS. Amazon S3 ne prend pas en charge les paramètres de blocage de l'accès public par objet.
- Lorsque vous appliquez des paramètres de blocage de l'accès public à un compte, ils s'appliquent à tous Régions AWS dans le monde entier. Les paramètres peuvent ne pas prendre effet immédiatement ou simultanément dans toutes les Régions, mais ils finissent par s'y propager.

Rubriques

Paramètres de la fonctionnalité de blocage de l'accès public

La fonctionnalité de blocage de l'accès public S3 fournit quatre paramètres. Vous pouvez appliquer ces paramètres de manière combinée à des points d'accès, à des compartiments ou à des Comptes AWS entiers. Si vous appliquez un paramètre à un compte, il s'applique à tous les compartiments et points d'accès appartenant à ce compte. De même, si vous appliquez un paramètre à un compartiment, il s'applique à tous les points d'accès associés à ce compartiment.

Le tableau suivant contient les paramètres disponibles.

Nom	Description
`BlockPublicAcls`	La configuration de cette option sur `TRUE` entraîne le comportement suivant : • Les appels PUT Bucket acl et PUT Object acl échouent si la liste de contrôle d'accès (ACL) spécifiée est publique. • Les appels PUT Object échouent si la demande inclut une ACL publique. • Si ce paramètre est appliqué à un compte, les appels PUT Bucket échouent si la demande inclut une ACL publique. Lorsque ce paramètre est défini sur`TRUE`, les opérations spécifiées échouent (qu'elles soient effectuées via l'API REST ou AWS les SDK). AWS CLI Toutefois, les stratégies et ACL existantes pour les compartiments et les objets ne sont pas modifiées. Ce paramètre vous protège contre l'accès public tout en vous permettant d'auditer, d'affiner ou de modifier les stratégies et ACL existantes pour vos compartiments et vos objets. **Note** Les points d'accès ne sont pas associés à des ACL. Si vous appliquez ce paramètre à un point d'accès, il agit comme une transmission au compartiment sous-jacent. Si ce paramètre est activé sur un point d'accès, les demandes effectuées via le point d'accès se comportent comme si ce paramètre était activé dans

Nom	Description
	le compartiment sous-jacent, que ce paramètre soit activé ou non dans le compartiment.
`IgnorePublicAcls`	La configuration de cette option sur TRUE conduit Amazon S3 à ignorer toutes les listes ACL publiques sur un compartiment et tout objet qu'il contient. Ce paramètre vous permet de bloquer en toute sécurité l'accès public accordé par les ACL, tout en autorisant les appels PUT Object qui incluent une ACL publique (contrairement à `BlockPublicAcls` qui rejettent les appels PUT Object incluant une ACL publique). L'activation de ce paramètre n'affecte pas la persistance des ACL existantes et n'empêche pas la configuration de nouvelles ACL publiques. **Note** Les points d'accès ne sont pas associés à des ACL. Si vous appliquez ce paramètre à un point d'accès, il agit comme une transmission au compartiment sous-jacent. Si ce paramètre est activé sur un point d'accès, les demandes effectuées via le point d'accès se comportent comme si ce paramètre était activé dans le compartiment sous-jacent, que ce paramètre soit activé ou non dans le compartiment.

Nom	Description
`BlockPublicPolicy`	En définissant cette option sur TRUE pour un compartiment, Amazon S3 rejette des appels vers une politique PUT Bucket si la politique de compartiment spécifiée autorise l'accès public. En définissant cette option sur TRUE pour un compartiment, Amazon S3 rejette également les appels à la politique de point d'accès PUT pour tous les points d'accès du même compte du compartiment si la politique spécifiée autorise l'accès public. Si vous définissez cette option sur TRUE pour un point d'accès, Amazon S3 rejette les appels à la stratégie PUT Access Point et à la stratégie PUT Bucket qui sont effectués via le point d'accès si la stratégie spécifiée (pour le point d'accès ou le compartiment sous-jacent) autorise l'accès public. Vous pouvez utiliser ce paramètre pour autoriser les utilisateurs à gérer des stratégies de point d'accès et de compartiment, sans toutefois les autoriser à partager publiquement le compartiment ou les objets qu'il contient. L'activation de ce paramètre n'a pas d'incidence sur les stratégies de point d'accès ou de compartiment existantes. **Important** Pour utiliser ce paramètre efficacement, nous vous recommandons de l'appliquer au niveau du compte. Une stratégie de compartiment peut autoriser les utilisateurs à modifier les paramètres de blocage de l'accès public d'un compte. Par conséquent, les utilisateurs autorisés à modifier une stratégie de compartiment peuvent insérer une stratégie qui leur permet de désactiver les paramètres de blocage de l'accès public pour le compartiment. Si ce paramètre est activé pour le compte entier, et non pour un compartiment spécifique, Amazon S3 bloque les stratégies publiques même si un utilisateur modifie la stratégie de compartiment pour désactiver ce paramètre.

Nom	Description
`RestrictPublicBuckets`	Si vous définissez cette option de `TRUE` manière à restreindre l'accès à un point d'accès ou à un bucket soumis à une politique publique, aux seuls responsables du AWS service et aux utilisateurs autorisés du compte du propriétaire du bucket et du compte du propriétaire du point d'accès. Ce paramètre bloque tous les accès entre comptes au point d'accès ou au compartiment (sauf pour les responsables du AWS service), tout en permettant aux utilisateurs du compte de gérer le point d'accès ou le compartiment. L'activation de ce paramètre n'a pas d'incidence sur les stratégies de point d'accès ou de compartiment existantes, mais Amazon S3 bloque l'accès public et entre comptes provenant des stratégies de point d'accès ou de compartiment publiques, notamment la délégation non publique à des comptes spécifiques.

Important

- Les appels vers GET Bucket acl et GET Object acl renvoient toujours les autorisations en vigueur pour le compartiment ou l'objet spécifié. Par exemple, supposons qu'un compartiment dispose d'une ACL accordant l'accès public, mais que le paramètre `IgnorePublicAcls` soit également activé pour le même compartiment. Dans ce cas, GET Bucket acl renvoie une ACL qui reflète les autorisations d'accès appliquées par Amazon S3, plutôt que l'ACL associée au compartiment.
- Les paramètres de blocage de l'accès public ne modifient pas les stratégies ou listes ACL existantes. Par conséquent, la suppression d'un paramètre de blocage de l'accès public conduit un compartiment ou un objet doté d'une ACL ou d'une stratégie publique à de nouveau être accessible publiquement.

Exécution d'opérations de blocage d'accès public sur un point d'accès

Pour effectuer des opérations de blocage de l'accès public sur un point d'accès, utilisez le AWS CLI services`s3control`.

> **Important**
>
> Notez qu'il n'est pas possible actuellement de modifier les paramètres de blocage d'accès public d'un point d'accès une fois ce dernier créé. Ainsi, la seule façon de spécifier les paramètres de blocage d'accès public pour un point d'accès est de les inclure lors de la création du point d'accès.

La signification du mot « public »

ACL

Amazon S3 considère qu'une ACL de compartiment ou d'objet est publique si elle accorde des autorisations aux membres des groupes `AllUsers` ou `AuthenticatedUsers` prédéfinis. Pour plus d'informations sur les groupes prédéfinis, consultez Groupes prédéfinis Amazon S3.

Stratégies de compartiment

Lors de l'évaluation d'une stratégie de compartiment, Amazon S3 commence par assumer que la stratégie est publique. Puis, il évalue la stratégie pour déterminer si elle qualifiée comme non publique. Pour être considérée comme non publique, une politique de compartiment doit accorder l'accès uniquement aux valeurs fixes (valeurs ne contenant aucun caractère générique ni aucune variable de politique AWS Identity and Access Management) d'un ou de plusieurs des éléments suivants :

- Un AWS principal, un utilisateur, un rôle ou un responsable de service (par exemple`aws:PrincipalOrgID`)
- Un ensemble de CIDR (Classless Inter-Domain Routing) utilisant `aws:SourceIp`. Pour plus d'informations sur CIDR, consultez RFC 4632 sur le site web RFC Editor.

> **Note**
>
> Les politiques de compartiment qui accordent un accès conditionné par la clé de condition `aws:SourceIp` avec de très larges plages d'adresses IP (par exemple, 0.0.0.0/1) sont considérées comme « publiques ». Cela inclut des valeurs supérieures à /8 pour IPv4 et à /32 pour IPv6 (à l'exception des plages privées RFC1918). Bloquer l'accès public rejette ces politiques « publiques » et empêche l'accès intercompte aux compartiments utilisant déjà ces politiques « publiques ».

- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `s3:x-amz-server-side-encryption-aws-kms-key-id`
- `aws:userid`, hors du modèle `"AROLEID:*"`
- `s3:DataAccessPointArn`

> **Note**
>
> Lorsqu'elle est utilisée dans une stratégie de compartiment, cette valeur peut contenir un caractère générique pour le nom du point d'accès sans rendre la stratégie publique, à condition que l'ID de compte soit corrigé. Par exemple, autoriser l'accès à `arn:aws:s3:us-west-2:123456789012:accesspoint/*` permettrait l'accès à n'importe quel point d'accès associé au compte `123456789012` dans la Région `us-west-2`, sans rendre la stratégie de compartiment publique. Notez que ce comportement est différent pour les stratégies de point d'accès. Pour plus d'informations, consultez Points d'accès.

- `s3:DataAccessPointAccount`

Pour plus d'informations sur les stratégies de compartiment, consultez Politiques relatives aux compartiments pour Amazon S3.

Example : politiques relatives aux compartiments publics

Conformément à ces règles, les exemples de stratégies suivants sont considérés comme publics.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
 }
```

```
{
```

```
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
 }
```

Ces stratégies peuvent devenir non publiques grâce à une valeur fixe en incluant l'une des clés de condition énumérée précédemment. Par exemple, la dernière stratégie ci-dessus peut devenir non publique si vous définissez `aws:SourceVpc` sur une valeur fixe, comme suit :

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
 }
```

Comment Amazon S3 évalue une politique de compartiment qui contient des autorisations d'accès publiques et non publiques ?

Cet exemple illustre comment Amazon S3 évalue une stratégie de compartiment contenant des autorisations d'accès publiques et non publiques.

Supposons qu'un compartiment dispose d'une stratégie qui accorde l'accès à un ensemble de mandataires fixes. Conformément aux règles précédemment décrites, cette stratégie n'est pas publique. Ainsi, si vous activez le paramètre `RestrictPublicBuckets`, la stratégie reste effective comme indiqué, car `RestrictPublicBuckets` s'applique uniquement aux compartiments disposant de stratégies publiques. Cependant, si vous ajoutez une déclaration publique à la stratégie, `RestrictPublicBuckets` s'applique au compartiment. Il permet uniquement aux responsables du AWS service et aux utilisateurs autorisés du compte du propriétaire du bucket d'accéder au bucket.

Par exemple, supposons qu'un compartiment détenu par « Account-1 » dispose d'une stratégie contenant les éléments suivants :

1. Une déclaration qui accorde l'accès à AWS CloudTrail (qui est un principal AWS de service)
2. Une instruction qui accorde l'accès au compte « Account-2 »
3. Une instruction qui accorde l'accès au public, par exemple en indiquant `"Principal": "*"` sans aucune `Condition` restrictive

Cette stratégie peut être publique à cause de la troisième instruction. Une fois cette politique en place et `RestrictPublicBuckets` activée, Amazon S3 autorise l'accès uniquement par CloudTrail. Notez que bien que l'instruction 2 ne soit pas publique, Amazon S3 désactive l'accès à « Account-2 ». En effet, l'instruction 3 permet à la stratégie entière de devenir publique, ainsi `RestrictPublicBuckets` s'applique. Par conséquent, Amazon S3 désactive l'accès entre comptes, bien que la stratégie délègue l'accès à un compte spécifique, « Account-2 ». Mais si vous supprimez l'instruction 3 de la stratégie, cette dernière ne peut pas être publique et `RestrictPublicBuckets` ne s'applique plus. Ainsi, « Account-2 » a de nouveau accès au compartiment, même si `RestrictPublicBuckets` reste activé.

Points d'accès

Amazon S3 évalue les paramètres de blocage de l'accès public de façon légèrement différente pour les points d'accès par rapport aux compartiments. Les règles qu'Amazon S3 applique pour déterminer quand une stratégie de point d'accès est publique sont généralement les mêmes pour les points d'accès et pour les compartiments, sauf dans les cas suivants :

- Un point d'accès ayant une origine réseau VPC est toujours considéré comme non public, quel que soit le contenu de sa stratégie de point d'accès.
- Une stratégie de point d'accès qui accorde l'accès à un ensemble de points d'accès utilisant `s3:DataAccessPointArn` est considérée comme publique. Notez que ce comportement est différent de celui des stratégies de compartiment. Par exemple, une stratégie de compartiment qui accorde l'accès aux valeurs de `s3:DataAccessPointArn` correspondant à `arn:aws:s3:us-west-2:123456789012:accesspoint/*` n'est pas considérée comme publique. Toutefois, la même instruction dans une stratégie de point d'accès rendrait le point d'accès public.

Utilisation de l'analyseur d'accès IAM pour S3 pour passer en revue les compartiments publics

Vous pouvez utiliser l'analyseur d'accès IAM pour S3 pour passer en revue les compartiments avec des listes ACL de compartiment, des politiques de compartiment ou des politiques de point d'accès qui accordent un accès public. IAM Access Analyzer for S3 vous avertit de la présence de compartiments configurés pour autoriser l'accès à toute personne sur Internet ou autre Comptes AWS, y compris Comptes AWS en dehors de votre organisation. Pour chaque compartiment public ou partagé, vous recevez des résultats qui signalent la source et le niveau d'accès public ou partagé.

Dans l'analyseur d'accès IAM pour S3, vous pouvez bloquer tout accès public à un compartiment en un seul clic. Vous pouvez également aller plus loin en configurant des niveaux d'accès précis dans les paramètres des niveaux d'autorisation des compartiments. Pour les cas d'utilisation spécifiques et vérifiés nécessitant un accès public ou partagé, vous pouvez confirmer et enregistrer votre intention de maintenir le niveau d'accès public ou partagé en archivant les résultats pour le compartiment.

Dans de rares cas, l'analyseur d'accès IAM pour S3 peut ne signaler aucun résultat pour un compartiment qu'une évaluation du blocage de l'accès public Amazon S3 signale comme public. Cela se produit parce que le blocage de l'accès public Amazon S3 examine les stratégies pour les actions en cours et les actions potentielles qui pourraient être ajoutées à l'avenir, ce qui rend un compartiment public. D'autre part, l'analyseur d'accès IAM pour S3 analyse uniquement les actions en cours spécifiées pour le service Amazon S3 dans l'évaluation du statut d'accès.

Pour plus d'informations sur l'analyseur d'accès IAM pour S3, consultez Examen de l'accès aux compartiments à l'aide de l'analyseur d'accès IAM pour S3.

Autorisations

Pour utiliser les fonctions du blocage de l'accès public Amazon S3, vous devez disposer des autorisations suivantes.

Opération	Autorisations requises
Statut de la stratégie GET Bucket	`s3:GetBucketPolicyStatus`
Paramètres de blocage de l'accès public GET bucket	`s3:GetBucketPublicAccessBlock`
Paramètres de blocage de l'accès public PUT bucket	`s3:PutBucketPublicAccessBlock`
Paramètres de blocage de l'accès public DELETE bucket	`s3:PutBucketPublicAccessBlock`
Paramètres de blocage de l'accès public GET account	`s3:GetAccountPublicAccessBlock`
Paramètres de blocage de l'accès public PUT account	`s3:PutAccountPublicAccessBlock`

Opération	Autorisations requises
Paramètres de blocage de l'accès public DELETE account	`s3:PutAccountPublicAccessBlock`
Paramètres de blocage de l'accès public PUT access point	`s3:CreateAccessPoint`

> **Note**
>
> Les opérations DELETE exigent les mêmes autorisations que les opérations PUT. Il n'existe pas d'autorisations séparées pour les opérations DELETE.

Configuration du blocage d'accès public

Pour plus d'informations sur la configuration de l'accès public par blocs pour vos compartiments Amazon S3 Compte AWS et ceux de votre compte Amazon S3, consultez les rubriques suivantes.

- Configuration des paramètres de blocage d'accès public pour votre compte
- Configuration des paramètres de blocage d'accès public pour vos compartiments S3

Configuration des paramètres de blocage d'accès public pour votre compte

La fonction du blocage de l'accès public Amazon S3 fournit des paramètres pour les points d'accès, les compartiments et les comptes afin de vous aider à gérer l'accès public aux ressources Amazon S3. Par défaut, les nouveaux compartiments, points d'accès et objets n'autorisent pas l'accès public.

Pour plus d’informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

> **Note**
>
> Les paramètres au niveau du compte remplacent les paramètres sur les objets individuels. La configuration de votre compte pour bloquer l'accès public annulera tous les paramètres d'accès public définis pour les objets individuels de votre compte.

Vous pouvez utiliser la console S3 AWS CLI, AWS les SDK et l'API REST pour configurer les paramètres de blocage de l'accès public pour tous les compartiments de votre compte. Consultez les sections ci-dessous pour en savoir plus.

Pour configurer les paramètres de blocage d'accès public pour vos compartiments, consultez Configuration des paramètres de blocage d'accès public pour vos compartiments S3. Pour plus d'informations sur les points d'accès, consultez Exécution d'opérations de blocage d'accès public sur un point d'accès.

Utiliser la console S3.

La fonctionnalité de blocage de l'accès public Amazon S3 empêche l'application de paramètres qui autorisent un accès public aux données dans des compartiments S3. Cette section explique comment modifier les paramètres de blocage de l'accès public pour tous les compartiments S3 de votre Compte AWS. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

Pour modifier les paramètres de blocage de l'accès public pour tous les compartiments S3 d'un Compte AWS

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Block Public Access settings for this account (Bloquer les paramètres d'accès public pour ce compte).
3. Choisissez Edit (Modifier) pour modifier les paramètres de blocage de l'accès public pour tous les compartiments de votre Compte AWS.
4. Choisissez les paramètres que vous souhaitez modifier, puis Save (Enregistrer).
5. Lorsque vous êtes invité à confirmer l'opération, entrez **`confirm`**. Choisissez ensuite Confirmer pour enregistrer vos modifications.

En utilisant le AWS CLI

Vous pouvez utiliser le blocage de l'accès public Amazon S3 via AWS CLI. Pour plus d'informations sur la configuration et l'utilisation du AWS CLI, voir Qu'est-ce que le AWS Command Line Interface ?

Compte

Pour effectuer des opérations Block Public Access sur un compte, utilisez le service AWS CLI de `s3control`. Les opérations au niveau des comptes qui utilisent ce service sont les suivantes :

- PUT PublicAccessBlock (pour un compte)
- GET PublicAccessBlock (pour un compte)
- SUPPRIMER PublicAccessBlock (pour un compte)

Pour des informations supplémentaires et des exemples, voir put-public-access-blockla AWS CLI référence.

Utilisation des AWS SDK

Java

Les exemples suivants vous montrent comment utiliser Amazon S3 Block Public Access AWS SDK for Java pour configurer un bloc d'accès public sur un compte Amazon S3.

```
AWSS3ControlClientBuilder controlClientBuilder =
 AWSS3ControlClientBuilder.standard();
controlClientBuilder.setRegion(<region>);
controlClientBuilder.setCredentials(<credentials>);

AWSS3Control client = controlClientBuilder.build();
client.putPublicAccessBlock(new PutPublicAccessBlockRequest()
  .withAccountId(<account-id>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withIgnorePublicAcls(<value>)
    .withBlockPublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

Important

Cet exemple s'applique uniquement aux opérations au niveau des comptes qui utilisent la classe client `AWSS3Control`. Pour les opérations au niveau des compartiments, consultez l'exemple précédent.

Other SDKs

Pour plus d'informations sur l'utilisation des autres AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Utilisation de l'API REST

Pour plus d’informations sur l’utilisation du blocage de l’accès public Amazon S3 via les API REST, consultez les rubriques suivantes dans la Référence d’API Amazon Simple Storage Service.

- Opérations au niveau des comptes
  - METTRE PublicAccessBlock
  - OBTENEZ PublicAccessBlock
  - SUPPRIMER PublicAccessBlock

Configuration des paramètres de blocage d'accès public pour vos compartiments S3

La fonction du blocage de l'accès public Amazon S3 fournit des paramètres pour les points d'accès, les compartiments et les comptes afin de vous aider à gérer l'accès public aux ressources Amazon S3. Par défaut, les nouveaux compartiments, points d'accès et objets n'autorisent pas l'accès public.

Pour plus d’informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Vous pouvez utiliser la console S3 AWS CLI, AWS les SDK et l'API REST pour accorder un accès public à un ou plusieurs compartiments. Vous pouvez également bloquer l'accès public à des compartiments qui sont déjà publics. Consultez les sections ci-dessous pour en savoir plus.

Pour configurer les paramètres de blocage de l'accès public pour chaque compartiment dans votre compte, consultez Configuration des paramètres de blocage d'accès public pour votre compte Pour plus d'informations sur la configuration du blocage de l'accès public des points d'accès, consultez Exécution d'opérations de blocage d'accès public sur un point d'accès.

Utiliser la console S3.

La fonctionnalité de blocage de l'accès public Amazon S3 empêche l'application de paramètres qui autorisent un accès public aux données dans des compartiments S3. Cette section explique comment modifier les paramètres de la fonctionnalité de blocage de l'accès public pour un ou plusieurs compartiments S3. Pour plus d'informations sur le blocage de l'accès public à l' AWS CLI aide AWS

des SDK et des API REST Amazon S3, consultezBlocage de l'accès public à votre stockage Amazon S3.

Vous pouvez voir si votre compartiment est accessible publiquement dans la liste Buckets (Compartiments). Dans la colonne Access (Accès), Amazon S3 étiquette les autorisations pour un compartiment comme suit :

- Public – N'importe qui a accès à une ou plusieurs des autorisations suivantes : Lister les objets, Écrire les objets, Lire et écrire des autorisations.
- Objects can be public (Les objets peuvent être publics) – Le compartiment n'est pas public, mais toute personne avec les autorisations appropriées peut accorder l'accès public à des objets.
- Bucket and objects not public (Compartiment et objets non publics) – Le compartiment et les objets n'ont aucun accès public.
- Uniquement les utilisateurs autorisés de ce compte : l'accès est limité aux utilisateurs et aux rôles IAM de ce compte et aux principaux responsables du AWS service, car il existe une politique qui accorde un accès public.

Vous pouvez également filtrer les recherches de compartiment par type d'accès. Choisissez un type d'accès dans la liste déroulante en regard de la barre Rechercher compartiments.

Si vous voyez`Error`lorsque vous listez vos compartiments et leurs paramètres d'accès public, il se peut que vous ne disposiez pas des autorisations requises. Assurez-vous d'avoir ajouté les autorisations suivantes à votre politique d'utilisateur ou de rôle :

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
s3:ListAllMyBuckets
```

Dans de rares cas, les demandes peuvent également échouer en raison d'un Région AWS Pannes.

Pour modifier les paramètres de blocage de l'accès public Amazon S3 pour un compartiment S3 simple

Suivez ces étapes si vous devez modifier les paramètres d'accès public pour un seul compartiment S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Nom du compartiment, choisissez le nom du compartiment.
3. Choisissez Permissions.
4. Choisissez Modifier pour modifier les paramètres d'accès public au compartiment. Pour en savoir plus sur les quatre paramètres de blocage de l'accès public Amazon S3, veuillez consulter Paramètres de la fonctionnalité de blocage de l'accès public.
5. Choisissez le paramètre que vous souhaitez modifier, puis Enregistrer.
6. Lorsque vous êtes invité à confirmer l'opération, entrez **`confirm`**. Choisissez ensuite Confirmer pour enregistrer vos modifications.

Vous pouvez modifier les paramètres de blocage de l'accès public Amazon S3 lorsque vous créez un compartiment. Pour plus d'informations, consultez Créer un compartiment.

En utilisant le AWS CLI

Pour bloquer l'accès public à un bucket ou pour supprimer le blocage d'accès public, utilisez le AWS CLI services`s3api`. Les opérations au niveau des compartiments qui utilisent ce service sont les suivantes :

- PUT PublicAccessBlock (pour un seau)
- GET PublicAccessBlock (pour un bucket)
- SUPPRIMER PublicAccessBlock (pour un bucket)
- OBTENEZ BucketPolicyStatus

Pour plus d'informations et des exemples, voir put-public-access-blockla AWS CLI référence.

Utilisation des AWS SDK

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
   .withCredentials(<credentials>)
   .build();
```

```
client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
  .withBucketName(<bucket-name>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withBlockPublicAcls(<value>)
    .withIgnorePublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

> **Important**
>
> Cet exemple s'applique uniquement aux opérations au niveau des compartiments qui utilisent la classe client `AmazonS3`. Pour les opérations au niveau des comptes, consultez l'exemple suivant.

Other SDKs

Pour plus d'informations sur l'utilisation des autres AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Utilisation de l'API REST

Pour plus d'informations sur l'utilisation du blocage de l'accès public Amazon S3 via les API REST, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service.

- Opérations au niveau des compartiments
  - METTRE PublicAccessBlock
  - OBTENEZ PublicAccessBlock
  - SUPPRIMER PublicAccessBlock
  - OBTENEZ BucketPolicyStatus

Examen de l'accès aux compartiments à l'aide de l'analyseur d'accès IAM pour S3

IAM Access Analyzer for S3 vous avertit de la présence de compartiments S3 configurés pour autoriser l'accès à toute personne sur Internet ou autre Comptes AWS, y compris Comptes AWS

en dehors de votre organisation. Pour chaque compartiment public ou partagé, vous recevez des résultats portant sur la source et le niveau des accès public ou partagé. Par exemple, l'analyseur d'accès IAM pour S3 peut montrer qu'un compartiment dispose d'un accès en lecture ou en écriture fourni via une liste de contrôle d'accès (ACL) de compartiment, une politique de compartiment, une politique de point d'accès multirégion ou une politique de point d'accès. Grâce à ces résultats, vous pouvez prendre des mesures correctives immédiates et précises pour rétablir l'accès à votre compartiment comme vous l'aviez prévu.

Lorsque vous examinez un compartiment à risque dans l'analyseur d'accès IAM pour S3, vous pouvez bloquer tout accès public au compartiment en un seul clic. Nous vous recommandons de bloquer tous les accès à vos compartiments, sauf si vous avez besoin d'un accès public pour prendre en charge un cas d'utilisation spécifique. Avant de bloquer tout accès public, assurez-vous que vos applications continueront à fonctionner correctement sans accès public. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Vous pouvez également aller plus loin en configurant des niveaux d'accès précis dans les paramètres des niveaux d'autorisation des compartiments. Pour des cas d'utilisation spécifiques et vérifiés nécessitant un accès public, tels que l'hébergement statique de site web, les téléchargements publics ou le partage entre comptes, vous pouvez confirmer et enregistrer votre intention pour que le compartiment reste public ou partagé en archivant les résultats du compartiment. Vous pouvez revisiter et modifier ces configurations de compartiments à tout moment. Vous pouvez également télécharger vos résultats sous forme de rapport CSV à des fins d'audit.

L'analyseur d'accès IAM pour S3 est disponible gratuitement sur la console Amazon S3. L'analyseur d'accès IAM pour S3 est optimisé par l'analyseur d'accès IAM d' AWS Identity and Access Management (IAM). Pour utiliser IAM Access Analyzer pour S3 dans la console Amazon S3, vous devez visiter la console IAM et activer IAM Access Analyzer par région.

Pour plus d'informations sur IAM Access Analyzer, voir Qu'est-ce qu'IAM Access Analyzer ? dans le guide de l'utilisateur IAM. Pour plus d'informations sur l'analyseur d'accès IAM pour S3, consultez les sections suivantes.

⚠ Important

- L'analyseur d'accès IAM pour S3 nécessite un analyseur au niveau du compte. Pour utiliser IAM Access Analyzer pour S3, vous devez visiter IAM Access Analyzer et créer un analyseur doté d'un compte comme zone de confiance. Pour plus d'informations, consultez Activation de l'analyseur d'accès IAM dans le Guide de l'utilisateur IAM.

- L'analyseur d'accès IAM pour S3 n'analyse pas la politique de point d'accès qui est attachée aux points d'accès intercomptes. Ce comportement se produit parce que le point d'accès et sa politique sont en dehors de la zone de confiance, c'est-à-dire du compte. Les compartiments qui délèguent l'accès à un point d'accès intercompte sont répertoriés sous Buckets with public access (Compartiments avec accès public) si vous n'avez pas appliqué le paramètre de blocage de l'accès public `RestrictPublicBuckets` au compartiment ou au compte. Lorsque vous appliquez le paramètre de `RestrictPublicBuckets` blocage de l'accès public, le compartiment est indiqué sous Compartiments accessibles par d'autres Comptes AWS , y compris des tiers Comptes AWS.
- Lorsqu'une politique de compartiment ou une liste de contrôle d'accès (ACL) de compartiment est ajoutée ou modifiée, l'analyseur d'accès IAM génère et met à jour les résultats en fonction de la modification dans un délai de 30 minutes. Il peut s'écouler jusqu'à six heures avant que les résultats relatifs aux paramètres de blocage de l'accès public au niveau du compte ne soient générés ou mis à jour après la modification des paramètres. Les résultats relatifs aux points d'accès multi-régions ne peuvent pas être générés ou mis à jour pendant six heures au maximum après la création, la suppression d'un point d'accès multi-régions ou la modification de sa politique.

Rubriques

Quelles informations l'analyseur d'accès IAM pour S3 fournit-il ?

L'analyseur d'accès IAM pour S3 fournit des résultats pour les compartiments accessibles hors de votre Compte AWS. Les compartiments répertoriés sous Compartiments avec accès public sont accessibles par n'importe quel utilisateur d'Internet. Si l'analyseur d'accès IAM pour S3 identifie des

compartiments publics, un avertissement en haut de la page indique le nombre de compartiments publics figurant dans votre région. Les compartiments répertoriés sous Compartiments accessibles depuis des tiers, y compris des tiers, Comptes AWS sont partagés conditionnellement avec d'autres personnes Comptes AWS, y compris des comptes extérieurs à votre organisation. Comptes AWS

Pour chaque compartiment, l'analyseur d'accès IAM pour S3 fournit les informations suivantes :

- Nom du compartiment
- Découvert par Access Analyzer : quand l'analyseur d'accès IAM pour S3 a découvert l'accès aux compartiments publics ou partagés.
- Shared through (Partagé via) – Mode de partage du compartiment : via une politique de compartiment, une liste de contrôle d'accès (ACL) ou une politique de point d'accès. Les points d'accès multi-régions et les points d'accès intercompte figurent sous la rubrique points d'accès. Un compartiment peut être partagé via des politiques et des listes de contrôle d'accès. Si vous souhaitez rechercher et examiner la source de votre accès au compartiment, vous pouvez utiliser les informations de cette colonne comme point de départ pour prendre des mesures correctives immédiates et précises.
- Status (Statut) – Statut du résultat de compartiment. L'analyseur d'accès IAM pour S3 affiche les résultats pour tous les compartiments publics et partagés.
  - Active (Actif) – Le résultat n'a pas été vérifié.
  - Archived (Archivé) – Le résultat a été vérifié et confirmé comme prévu.
  - Tous - Tous les résultats relatifs à des buckets publics ou partagés avec d'autres personnes Comptes AWS, y compris Comptes AWS en dehors de votre organisation.
- Access level (Niveau d'accès) – Autorisations d'accès accordées pour le compartiment :
  - List (Liste) – Répertorier les ressources.
  - Read (Lecture) – Lire mais ne pas modifier les contenus et attributs de ressources.
  - Write (Écriture) – Créer, supprimer ou modifier des ressources.
  - Permissions (Autorisations) – Accorder ou modifier des autorisations de ressources.
  - Tagging (Balisage) – Mettre à jour les balises associées à la ressource.

Activation de l'analyseur d'accès IAM pour S3

Pour utiliser l'analyseur d'accès IAM pour S3, vous devez effectuer les étapes prérequises suivantes.

1. Accordez les autorisations requises.

Pour plus d'informations, consultez Autorisations requises pour l'utilisation de l'analyseur d'accès IAM dans le Guide de l'utilisateur IAM.

2. Accédez à IAM pour créer un analyseur au niveau du compte pour chaque région où vous souhaitez utiliser l'analyseur d'accès IAM.

   L'analyseur d'accès IAM pour S3 nécessite un analyseur au niveau du compte. Pour utiliser l'analyseur d'accès IAM pour S3, vous devez créer un analyseur doté d'un compte comme zone de confiance. Pour plus d'informations, consultez Activation de l'analyseur d'accès IAM dans le Guide de l'utilisateur IAM.

Blocage de tous les accès publics

Si vous souhaitez bloquer tout accès à un compartiment en un seul clic, vous pouvez utiliser le bouton Bloquer tout l'accès public dans l'analyseur d'accès IAM pour S3. Lorsque vous bloquez tout accès public à un compartiment, aucun accès public n'est accordé. Nous vous recommandons de bloquer tous les accès publics à vos compartiments, sauf si vous avez besoin d'un accès public pour prendre en charge un cas d'utilisation spécifique et vérifié. Avant de bloquer tout accès public, assurez-vous que vos applications continueront à fonctionner correctement sans accès public.

Si vous ne souhaitez pas bloquer tous les accès publics à votre compartiment, vous pouvez modifier vos paramètres de blocage de l'accès public dans la console Amazon S3 pour configurer des niveaux d'accès précis à vos compartiments. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Dans de rares cas, l'analyseur d'accès IAM pour S3 peut ne signaler aucun résultat pour un compartiment qu'une évaluation du blocage de l'accès public Amazon S3 signale comme public. Cela se produit parce que le blocage de l'accès public Amazon S3 examine les stratégies pour les actions en cours et les actions potentielles qui pourraient être ajoutées à l'avenir, ce qui rend un compartiment public. D'autre part, l'analyseur d'accès IAM pour S3 analyse uniquement les actions en cours spécifiées pour le service Amazon S3 dans l'évaluation du statut d'accès.

Pour bloquer tout accès public à un compartiment à l'aide de l'analyseur d'accès IAM pour S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, sous Dashboards (Tableaux de bord), choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).

3. Dans l'analyseur d'accès IAM pour S3, choisissez un compartiment.
4. Choisissez Block all public access (Bloquer tous les accès publics).
5. Pour confirmer votre intention de bloquer tout accès public au compartiment, dans Block all public access (bucket settings) (Bloquer tout accès public (paramètres du compartiment)), entrez **`confirm`**.

   Amazon S3 bloque tout accès public à votre compartiment. Le statut du résultat de compartiment devient résolu et le compartiment disparaît de la liste de l'analyseur d'accès IAM pour S3. Si vous souhaitez consulter les buckets résolus, ouvrez IAM Access Analyzer sur la console IAM.

Vérification et modification de l'accès à un compartiment

Si vous n'aviez pas l'intention d'accorder l'accès au public ou à d'autres Comptes AWS personnes, y compris à des comptes extérieurs à votre organisation, vous pouvez modifier l'ACL du bucket, la politique du bucket, la politique du point d'accès multirégional ou la politique du point d'accès pour supprimer l'accès au bucket. La colonne Shared through (Partagé via) affiche toutes les sources d'accès au compartiment : politique de compartiment, ACL de compartiment et/ou politique de point d'accès. Les points d'accès multi-régions et les points d'accès intercompte figurent sous la rubrique points d'accès.

Pour consulter et modifier une politique de compartiment, une liste ACL de compartiment ou une politique de point d'accès

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).
3. Pour déterminer si l'accès public ou l'accès partagé est accordé via une politique de compartiment, une liste de contrôle d'accès de compartiment ou une politique de point d'accès, consultez la colonne Shared through (Partagé via).
4. Sous Buckets (Compartiments), choisissez le nom du compartiment avec la politique de compartiment, la liste ACL de compartiment, la politique de point d'accès multi-Régions ou la politique de point d'accès que vous souhaitez modifier ou vérifier.
5. Si vous souhaitez modifier ou consulter la liste ACL d'un compartiment :

   a. Choisissez Permissions.
   b. Choisissez Access Control List.
   c. Consultez la liste ACL de votre compartiment et apportez les modifications nécessaires.

Pour plus d’informations, consultez Configuration des listes ACL.

6. Si vous souhaitez modifier ou vérifier une politique de compartiment :

   a. Choisissez Permissions.
   b. Choisissez Stratégie de compartiment.
   c. Vérifiez ou modifiez votre politique de compartiment selon vos besoins.

      Pour plus d’informations, consultez Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3.

7. Si vous souhaitez consulter ou modifier une politique de point d'accès :

   a. ChoisissezPoint d'accès multi-Régions.
   b. Choisissez le nom du point d'accès multi-Régions.
   c. Vérifiez ou modifiez votre politique de point d'accès multi-Régions selon vos besoins.

      Pour plus d’informations, consultez Autorisations.

8. Si vous souhaitez consulter ou modifier une politique de point d'accès :

   a. Choisissez Access Points (Points d'accès).
   b. Choisissez le nom du point d'accès.
   c. Vérifiez ou modifiez l'accès en fonction de vos besoins.

      Pour plus d’informations, consultez Utilisation des points d'accès Amazon S3 dans la console Amazon S3.

   Si vous modifiez ou supprimez une liste ACL de compartiment, une politique de compartiment ou une politique de point d'accès pour supprimer l'accès public ou partagé, le statut du compartiment devient « résolu ». Les résultats du bucket résolus disparaissent de la liste IAM Access Analyzer pour S3, mais vous pouvez les consulter dans IAM Access Analyzer.

Archivage des résultats de compartiment

Si un bucket accorde l'accès au public ou à d'autres personnes Comptes AWS, y compris à des comptes extérieurs à votre organisation, pour prendre en charge un cas d'utilisation spécifique (par exemple, un site Web statique, des téléchargements publics ou le partage entre comptes), vous pouvez archiver les résultats du bucket. Lorsque vous archivez les résultats d'un compartiment, vous

confirmez et enregistrez votre intention de le garder public ou partagé. Les résultats de compartiment archivés restent dans votre liste de l'analyseur d'accès IAM pour S3 pour que vous sachiez toujours quels compartiments sont publics ou partagés.

Pour archiver les résultats de compartiment dans l'analyseur d'accès IAM pour S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).
3. Dans l'analyseur d'accès IAM pour S3, choisissez un compartiment actif.
4. Pour confirmer votre intention de rendre ce compartiment accessible au public ou à d'autres personnes Comptes AWS, y compris à des comptes extérieurs à votre organisation, choisissez Archiver.
5. Entrez **`confirm`**, puis choisissez Archive (Archiver).

Activation d'un résultat de compartiment archivé

Après avoir archivé des résultats, vous pouvez toujours les revoir et faire passer leur statut à « actif », ce qui indique que le compartiment nécessite une autre vérification.

Pour activer un résultat de compartiment archivé dans l'analyseur d'accès IAM pour S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).
3. Choisissez les résultats de compartiment archivés.
4. Choisissez Mark as active (Marquer comme actif).

Affichage des détails de résultats

Si vous avez besoin de plus d'informations sur un bucket, vous pouvez ouvrir les informations relatives à la recherche du bucket dans IAM Access Analyzer sur la console IAM.

Pour afficher les détails des résultats dans l'analyseur d'accès IAM pour S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).
3. Dans l'analyseur d'accès IAM pour S3, choisissez un compartiment.

4. Choisissez Afficher les détails.

   Les détails de la recherche s'affichent dans IAM Access Analyzer sur la console IAM.

Téléchargement d'un rapport de l'analyseur d'accès IAM pour S3

Vous pouvez télécharger vos résultats de compartiment sous la forme d'un rapport CSV que vous pouvez utiliser à des fins d'audit. Ce rapport inclut les mêmes informations que celles que vous voyez dans l'analyseur d'accès IAM pour S3 dans la console Amazon S3.

Pour télécharger un rapport

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation de gauche, choisissez Access Analyzer for S3 (Analyseur d'accès pour S3).
3. Dans le filtre de Région, sélectionnez la Région.

   L'analyseur d'accès IAM pour S3 est mis à jour pour afficher les compartiments pour la région choisie.
4. Choisissez Download report (Télécharger le rapport).

   Un rapport CSV est généré et enregistré sur votre ordinateur.

Vérification de la propriété du compartiment avec la condition du propriétaire du compartiment

La condition du propriétaire du compartiment Amazon S3 garantit que les compartiments que vous utilisez dans vos opérations S3 correspondent à vos attentes. Comptes AWS

La plupart des opérations S3 lisent ou écrivent dans des compartiments S3 spécifiques. Ces opérations incluent le chargement, la copie et le téléchargement d'objets, la récupération ou la modification de configurations de compartiments et la récupération ou la modification de configurations d'objets. Lorsque vous exécutez ces opérations, vous spécifiez le compartiment que vous souhaitez utiliser en incluant son nom avec la demande. Par exemple, pour récupérer un objet à partir de S3, vous effectuez une demande qui spécifie le nom d'un compartiment et la clé d'objet à extraire depuis ce compartiment.

Étant donné qu'Amazon S3 identifie les compartiments en fonction de leurs noms, une application qui utilise un nom de compartiment incorrect dans une demande peut effectuer par inadvertance des opérations sur un compartiment différent de ce qui était prévu. Pour éviter les interactions de compartiments involontaires dans des situations comme celle-ci, vous pouvez utiliser la condition propriétaire du compartiment. La condition propriétaire du compartiment vous permet de vérifier que le compartiment cible appartient au Compte AWS prévu, ce qui vous offre une garantie supplémentaire que vos opérations S3 ont les effets souhaités.

Rubriques

Quand utiliser la condition propriétaire du compartiment

Nous vous recommandons d'utiliser la condition propriétaire du compartiment chaque fois que vous exécutez une opération S3 prise en charge et que vous connaissez l'ID de compte du propriétaire du compartiment prévu. La condition propriétaire du compartiment est disponible pour toutes les opérations d'objet S3 et la plupart des opérations de compartiment S3. Pour obtenir la liste des opérations S3 ne prenant pas en charge la condition propriétaire du compartiment, veuillez consulter Limites et restrictions.

Pour voir les avantages de l'utilisation de la condition du propriétaire du bucket, imaginez le scénario suivant impliquant le AWS client Bea :

1. Bea développe une application qui utilise Amazon S3. Pendant le développement, Bea utilise ses tests uniquement Compte AWS pour créer un bucket nommé`bea-data-test`, et configure son application pour qu'elle y envoie des demandes. `bea-data-test`
2. Bea déploie son application, mais oublie de reconfigurer l'application pour qu'elle utilise un compartiment dans son Compte AWS de production.
3. En production, l'application de Bea effectue des demandes réussies à `bea-data-test` Cela entraîne l'écriture de données de production dans le compartiment du compte de test de Bea.

Bea peut se protéger contre les situations de ce genre en utilisant la condition propriétaire du compartiment. Avec l'état du propriétaire du bucket, Bea peut inclure l' Compte AWS identifiant du

propriétaire attendu du bucket dans ses demandes. Amazon S3 vérifie ensuite l'ID de compte du propriétaire du compartiment avant de traiter chaque demande. Si le propriétaire du compartiment réel ne correspond pas au propriétaire du compartiment prévu, la demande échoue.

Si Bea utilise la condition propriétaire du compartiment, le scénario décrit précédemment n'entraînera pas l'écriture par inadvertance de l'application de Bea dans un compartiment du compte de test. Au lieu de cela, les demandes effectuées par l'application lors de l'étape 3 échoueront avec un message d'erreur `Access Denied`. En utilisant la condition propriétaire du compartiment, Bea élimine le risque d'interaction accidentelle avec des compartiments dans le mauvais Compte AWS.

Vérification d'un propriétaire du compartiment

Pour utiliser la condition propriétaire du compartiment, vous incluez à votre demande un paramètre qui spécifie le propriétaire du compartiment prévu. La plupart des opérations S3 impliquent seulement un compartiment unique et n'exigent que ce paramètre unique pour utiliser la condition propriétaire du compartiment. Pour les opérations `CopyObject`, ce premier paramètre spécifie le propriétaire prévu du compartiment de destination et vous incluez un second paramètre pour spécifier le propriétaire prévu du compartiment source.

Lorsque vous effectuez une demande qui inclut un paramètre de condition propriétaire du compartiment, S3 compare l'ID de compte du propriétaire du compartiment et le paramètre spécifié avant de traiter la demande. Si le paramètre correspond à l'ID de compte du propriétaire du compartiment, S3 traite la demande. Si le paramètre ne correspond pas à l'ID de compte du propriétaire du compartiment, la demande échoue avec un message d'erreur `Access Denied`.

Vous pouvez utiliser la condition du propriétaire du compartiment avec les AWS Command Line Interface (AWS CLI), AWS les SDK et les API REST Amazon S3. Lorsque vous utilisez la condition du propriétaire du compartiment avec les API REST AWS CLI et Amazon S3, utilisez les noms de paramètres suivants.

Méthode d'accès	Paramètre pour les opérations non copiées	Paramètre source de l'opération Copy	Paramètre de destination de l'opération Copy
AWS CLI	`--expected-bucket-owner`	`--expected-source-bucket-owner`	`--expected-bucket-owner`

Méthode d'accès	Paramètre pour les opérations non copiées	Paramètre source de l'opération Copy	Paramètre de destination de l'opération Copy
API REST Amazon S3	En-tête `x-amz-expected-bucket-owner`	En-tête `x-amz-source-expected-bucket-owner`	En-tête `x-amz-expected-bucket-owner`

Les noms des paramètres requis pour utiliser la condition propriétaire du compartiment avec les kits AWS SDK varient en fonction de la langue. Pour déterminer les paramètres requis, consultez la documentation du kit SDK correspondant à la langue souhaitée. Vous trouverez la documentation du kit SDK dans Outils pour créer sur AWS.

Exemples

Les exemples suivants montrent comment implémenter la condition du propriétaire du compartiment dans Amazon S3 à l'aide du AWS CLI ou du AWS SDK for Java 2.x.

Example

Exemple : Chargement d'un objet

L'exemple suivant charge un objet dans le compartiment S3 *example-s3-bucket1* à l'aide de la condition propriétaire du compartiment pour s'assurer que *example-s3-bucket1* appartient au Compte AWS 111122223333.

AWS CLI

```
aws s3api put-object \
                --bucket example-s3-bucket1 --key exampleobject --
body example_file.txt \
                --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();;
    PutObjectRequest request = PutObjectRequest.builder()
            .bucket("example-s3-bucket1")
```

```
            .key("exampleobject")
            .expectedBucketOwner("111122223333")
            .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Exemple : Copie d'un objet

L'exemple suivant copie l'objet `object1` à partir du compartiment S3 *example-s3-bucket1* vers le compartiment S3 *example-s3-bucket2*. Il utilise la condition propriétaire du compartiment pour s'assurer que les compartiments appartiennent aux comptes prévus en fonction du tableau suivant.

Compartiment	Propriétaire prévu
example-s3-bucket1	111122223333
example-s3-bucket2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source example-s3-bucket1/object1 \
                            --bucket example-s3-bucket2 --key object1copy \
                            --expected-source-bucket-owner 111122223333 --expected-
bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
        S3Client s3Client = S3Client.create();
        CopyObjectRequest request = CopyObjectRequest.builder()
                .copySource("example-s3-bucket1/object1")
                .destinationBucket("example-s3-bucket2")
                .destinationKey("object1copy")
                .expectedSourceBucketOwner("111122223333")
                .expectedBucketOwner("444455556666")
                .build();
        s3Client.copyObject(request);
```

```
    }
```

Example

Exemple : Récupération d'une stratégie de compartiment

L'exemple suivant récupère la stratégie d'accès pour le compartiment S3 *example-s3-bucket1* à l'aide de la condition propriétaire du compartiment pour s'assurer que *example-s3-bucket1* appartient au Compte AWS 111122223333.

AWS CLI

```
aws s3api get-bucket-policy --bucket example-s3-bucket1 --expected-bucket-
owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
            .bucket("example-s3-bucket1")
            .expectedBucketOwner("111122223333")
            .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Limites et restrictions

La condition propriétaire du compartiment Amazon S3 comporte les restrictions et limitations suivantes :

- La valeur du paramètre de condition du propriétaire du compartiment doit être un Compte AWS ID (valeur numérique à 12 chiffres). Les principals de service ne sont pas pris en charge.

- La condition de propriétaire du bucket n'est pas disponible pour CreateBucketou aucune des opérations incluses dans AWS S3 Control. ListBuckets Amazon S3 ignore tous les paramètres de condition propriétaire du compartiment inclus dans les demandes envoyées à ces opérations.
- La condition propriétaire du compartiment vérifie uniquement que le le compartiment appartient au compte spécifié dans le paramètre de vérification. La condition propriétaire du compartiment ne vérifie pas la configuration du compartiment. Elle ne garantit pas non plus que la configuration du compartiment remplit des conditions spécifiques ou correspond à un état antérieur.

Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes de contrôle d'accès (ACL). Par défaut, la propriété d'objets est définie sur le paramètre Propriétaire du compartiment appliqué, et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets présents dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL ; nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès pour chaque objet individuellement. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler plus facilement l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment.

Object Ownership (Propriété de l'objet) dispose de trois paramètres que vous pouvez utiliser pour contrôler la propriété des objets téléchargés dans votre compartiment pour désactiver ou activer les listes ACL :

Listes ACL désactivées

- Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations sur les données du compartiment S3. Le compartiment utilise des stratégies pour définir le contrôle des accès.

Listes ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.
- Object writer (Rédacteur d'objets) – Le Compte AWS qui télécharge un objet est propriétaire de l'objet, a un contrôle total sur celui-ci et peut en accorder l'accès à d'autres utilisateurs via des listes ACL.

Pour la majorité des cas d'utilisation modernes dans S3, nous vous recommandons de maintenir les listes ACL désactivées en appliquant le paramètre Propriétaire du compartiment appliqué et en utilisant votre politique de compartiment pour partager des données avec des utilisateurs extérieurs à votre compte, selon les besoins. Cette approche simplifie la gestion des autorisations. Vous pouvez désactiver les listes de contrôle d'accès sur les compartiments nouvellement créés et déjà existants. Pour les compartiments nouvellement créés, les listes ACL sont désactivées par défaut. Dans le cas d'un compartiment existant qui contient déjà des objets, une fois que vous avez désactivé les ACL, les listes ACL d'objet et de compartiment ne font plus partie d'une évaluation d'accès et l'accès est accordé ou refusé sur la base de stratégies. Pour les compartiments existants, vous pouvez réactiver les ACL à tout moment après les avoir désactivées, et vos listes ACL de compartiment et d'objets préexistantes sont restaurées.

Avant de désactiver les listes ACL, nous vous recommandons de revoir votre stratégie de compartiment pour vous assurer qu'elle couvre toutes les façons dont vous avez l'intention d'accorder l'accès à votre compartiment hors de votre compte. Une fois que vous avez désactivé les listes ACL, votre compartiment accepte uniquement les requêtes PUT qui ne spécifient pas de requête ACL ou PUT avec des listes ACL de contrôle total du propriétaire du compartiment, telles que la liste ACL `bucket-owner-full-control` prédéfinie ou des formes équivalentes de cette ACL exprimées en XML. Les applications existantes qui prennent en charge les ACL de contrôle total du propriétaire du bucket n'ont aucun impact. PUTles demandes contenant d'autres ACL (par exemple, des autorisations personnalisées accordées à certaines Comptes AWS) échouent et renvoient une `400` erreur avec le code `AccessControlListNotSupported` d'erreur.

Au contraire, un compartiment avec le paramètre Propriétaire du compartiment préféré continue d'accepter et d'honorer les listes ACL de compartiment et d'objet. Avec ce paramètre, de nouveaux objets écrits avec la liste ACL `bucket-owner-full-control` prédéfinie sont automatiquement détenus par le propriétaire du compartiment plutôt que par l'auteur d'objets. Tous les autres comportements ACL restent en place. Pour exiger que toutes les opérations PUT Amazon S3 incluent

la liste ACL `bucket-owner-full-control` prédéfinie, vous pouvez ajouter une politique de compartiment qui n'autorise que les chargements d'objets à l'aide de cette ACL.

Pour voir quels paramètres de propriété d'objets sont appliqués à vos compartiments, vous pouvez utiliser les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Using S3 Storage Lens to find Object Ownership settings (Utilisation de S3 Storage Lens pour trouver les paramètres de propriété des objets).

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Paramètres de la propriété de l'objet

Ce tableau montre l'impact de chaque paramètre de la propriété d'objet sur les listes ACL, les objets, la propriété des objets et les téléchargements d'objets.

Paramètre	S'applique à	Effet sur la propriété de l'objet	Effet sur les listes ACL	Chargements acceptés
Propriétaire du compartiment appliqué (par défaut)	Tous les objets existants et nouveaux	Le propriétaire du compartiment est propriétaire de chaque objet.	Les listes ACL sont désactivées et n'affectent plus les autorisations d'accès à votre compartiment. Les demandes de définition ou de mise à jour des listes ACL échouent.	Chargements avec des listes ACL de contrôle total du propriétaire du compartiment ou des téléchargements qui ne spécifient pas de liste ACL

Paramètre	S’applique à	Effet sur la propriété de l'objet	Effet sur les listes ACL	Chargements acceptés
			Cependant, les demandes de lecture de listes ACL sont prises en charge. Le propriétaire du compartiment possède la propriété et le contrôle complets. Le rédacteur d'objets n'a plus la propriété et le contrôle complets.	

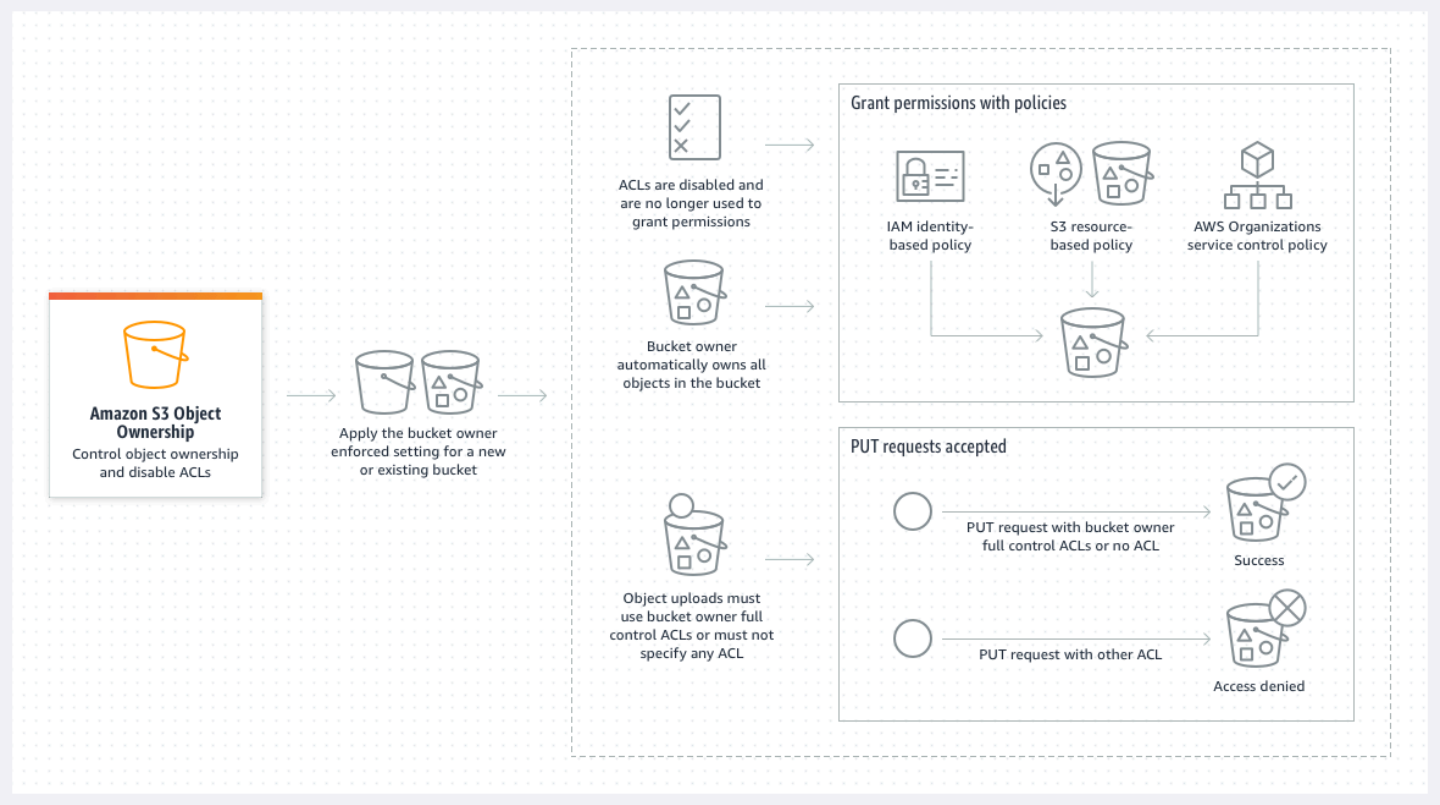
Paramètre	S’applique à	Effet sur la propriété de l'objet	Effet sur les listes ACL	Chargements acceptés
Propriétaire du compartiment préféré	Nouveaux objets	Si le chargement d'un objet inclut la liste ACL `bucket-owner-full-control` prédéfinie, le propriétaire du compartiment est propriétaire de l'objet. Les objets téléchargés avec d'autres listes ACL appartiennent au compte d'écriture.	Les listes ACL peuvent être mises à jour et peuvent accorder des autorisations. Si le chargement d'un objet inclut la liste ACL `bucket-owner-full-control` prédéfinie, le propriétaire du compartiment dispose d'un accès en contrôle total et l'enregistreur d'objets n'a plus d'accès de contrôle complet.	Tous les chargements

Paramètre	S'applique à	Effet sur la propriété de l'objet	Effet sur les listes ACL	Chargements acceptés
Créateur d'objets	Nouveaux objets	Le rédacteur d'objet possède l'objet.	Les listes ACL peuvent être mises à jour et peuvent accorder des autorisations. Le rédacteur d'objet dispose d'un accès complet.	Tous les chargements

Les changements introduits par la désactivation des listes ACL

Lorsque vous appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, les listes ACL sont désactivées et vous possédez automatiquement tous les objets du compartiment et en prenez le contrôle total sans effectuer d'actions supplémentaires. Propriétaire du compartiment appliqué est le paramètre par défaut pour tous les compartiments nouvellement créés. Après l'application du paramètre Propriétaire du compartiment appliqué, trois changements sont notables :

- Toutes les listes ACL de compartiment et les listes ACL d'objets sont désactivées, ce qui vous donne un accès complet, en tant que propriétaire du compartiment. Lorsque vous exécutez une demande ACL en lecture sur votre compartiment ou votre objet, vous verrez que l'accès complet n'est accordé qu'au propriétaire du compartiment.
- Avec ce paramètre, en tant que propriétaire du compartiment, vous possédez automatiquement tous les objets de votre compartiment.
- Les listes ACL n'affectent plus les autorisations d'accès à votre compartiment. Par conséquent, le contrôle d'accès à vos données est basé sur des stratégies, telles que les stratégies IAM, les stratégies de compartiment S3, les stratégies de point de terminaison d'un VPC et les politiques de contrôle des services (SCP) des organisations.



Si vous utilisez la gestion des versions S3, le propriétaire du compartiment possède et contrôle total sur toutes les versions d'objets de votre compartiment. L'application du paramètre Propriétaire du compartiment appliqué n'ajoute pas de nouvelle version d'un objet.

Les nouveaux objets peuvent être chargés dans votre compartiment uniquement s'ils utilisent des listes ACL de contrôle total du propriétaire du compartiment ou ne spécifient pas de liste ACL. Les téléchargements d'objets échouent s'ils spécifient une autre liste ACL. Pour plus d'informations, consultez Résolution des problèmes.

Étant donné que l'exemple suivant de l'opération `PutObject` utilise à l'aide de l' AWS Command Line Interface (AWS CLI) inclut la liste ACL `bucket-owner-full-control`prédéfinie, l'objet peut être chargé dans un compartiment avec des listes ACL désactivées.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file --
acl bucket-owner-full-control
```

Étant donné que l'opération `PutObject` suivante ne spécifie pas une liste ACL, elle réussit également pour un compartiment avec des listes ACL désactivées.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key key-name --body path-to-file
```

> **Note**
>
> Si d'autres utilisateurs Comptes AWS ont besoin d'accéder aux objets après le téléchargement, vous devez accorder des autorisations supplémentaires à ces comptes par le biais de politiques relatives aux compartiments. Pour plus d'informations, consultez Procédures pas à pas utilisant des politiques pour gérer l'accès à vos ressources Amazon S3.

Réactivation des listes ACL

Vous pouvez réactiver les listes ACL en passant du paramètre Propriétaire du compartiment appliqué à un autre paramètre de propriété d'objets à tout moment. Si vous avez utilisé des listes ACL d'objet pour la gestion des autorisations avant d'appliquer le paramètre Propriétaire du compartiment appliqué et que vous n'avez pas migré ces autorisations ACL d'objet vers votre politique de compartiment, après avoir réactivé les listes ACL, ces autorisations sont restaurées. De plus, les objets écrits dans le compartiment pendant que le paramètre Propriétaire du compartiment appliqué était appliqué appartiennent toujours au propriétaire du compartiment.

Par exemple, si du paramètre Propriétaire du compartiment appliqué vous revenez au paramètre Créateur d'objets, en tant que propriétaire du compartiment, vous ne possédez plus les objets qui appartenaient auparavant à d'autres Comptes AWS, et n'en avez plus le contrôle total. Au lieu de cela, les comptes de chargement sont à nouveau propriétaires de ces objets. Les objets appartenant à d'autres comptes utilisent des listes ACL pour les autorisations. Vous ne pouvez donc pas utiliser de stratégies pour accorder des autorisations à ces objets. Toutefois, en tant que propriétaire du compartiment, vous possédez toujours les objets qui ont été écrits dans le compartiment quand le paramètre Propriétaire du compartiment appliqué était appliqué. Ces objets ne sont pas détenus par le rédacteur d'objet, même si vous réactivez les listes ACL.

Pour obtenir des instructions sur l'activation et la gestion des ACL à l'aide de l' AWS Management Console interface de ligne de commande AWS Command Line Interface (CLI), de l'API REST ou AWS des SDK, consultez. Configuration des listes ACL

Conditions préalables à la désactivation des listes ACL

Avant de désactiver les listes ACL pour un compartiment existant, remplissez les prérequis suivants.

Examinez les listes ACL de compartiment et d'objet et migrez les autorisations ACL

Lorsque vous désactivez les listes ACL, les autorisations accordées par les listes ACL de compartiment et d'objets n'affectent plus l'accès. Avant de désactiver les listes ACL, vérifiez les listes ACL de votre compartiment et de vos objets.

Si vos listes ACL de compartiment accordent des autorisations de lecture ou d'écriture à d'autres personnes hors de votre compte, vous devez migrer ces autorisations vers votre politique de compartiment avant de pouvoir appliquer le paramètre Propriétaire du compartiment appliqué. Si vous ne migrez pas les listes ACL de compartiment qui accordent un accès en lecture ou en écriture hors de votre compte, votre demande d'application du paramètre Propriétaire du compartiment appliqué échoue et renvoie le code d'erreur InvalidBucketAclWithObjectOwnership.

Par exemple, si vous souhaitez désactiver les listes ACL pour un compartiment qui reçoit les journaux d'accès au serveur, vous devez migrer les autorisations ACL du compartiment pour le groupe de mise à disposition des journaux S3 vers le principal du service de journalisation dans une stratégie de compartiment. Pour plus d'informations, consultez Octroi de l'accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur.

Si vous souhaitez que le rédacteur d'objets conserve le contrôle total de l'objet qu'il télécharge, le rédacteur d'objets est le meilleur paramètre de propriété d'objets pour votre cas d'utilisation. Si vous souhaitez contrôler l'accès au niveau de l'objet individuel, le propriétaire du compartiment préféré est le meilleur choix. Ces cas d'utilisation sont rares.

Pour consulter les listes ACL et migrer les autorisations ACL vers des stratégies de compartiment, consultez la section Conditions préalables à la désactivation des listes ACL.

Identifier toutes les demandes qui ont nécessité une liste ACL pour l'autorisation

Pour identifier les demandes Amazon S3 qui ont nécessité des listes ACL pour l'autorisation, vous pouvez utiliser la valeur `aclRequired` dans les journaux d'accès du serveur Amazon S3 ou AWS CloudTrail. Si la demande a nécessité une liste ACL pour l'autorisation ou si vous avez des demandes PUT qui spécifient une liste ACL, la chaîne est `Yes`. Si aucune ACL n'est requise, ou si vous définissez une ACL `bucket-owner-full-control` prédéfinie, ou si les demandes sont autorisées par votre politique de compartiment, la chaîne de `aclRequired` valeur est « `-` » dans les journaux d'accès au serveur Amazon S3 et est absente dans CloudTrail. Pour plus d'informations sur les valeurs `aclRequired` attendues, consultez Valeurs `aclRequired` pour les demandes Amazon S3 courantes.

Si vous avez des demandes `PutBucketAcl` et `PutObjectAcl` contenant des en-têtes qui accordent des autorisations basées sur des listes ACL, à l'exception de la liste ACL `bucket-owner-full-control` prédéfinie, vous devez supprimer ces en-têtes avant de pouvoir désactiver les listes ACL. Dans le cas contraire, vos demandes échoueront.

Pour toutes les autres demandes nécessitant une liste ACL pour l'autorisation, migrez ces autorisations de liste ACL vers des politiques de compartiment. Supprimez ensuite toutes les listes ACL du compartiment avant d'activer le paramètre appliqué par le propriétaire du compartiment.

> **Note**
>
> Ne supprimez pas les listes ACL d'objet. Sinon, les applications qui s'appuient sur des listes ACL d'objet pour les autorisations perdent l'accès.

Si vous constatez qu'aucune demande ne nécessite une liste ACL pour l'autorisation, vous pouvez procéder à la désactivation des listes ACL. Pour plus d'informations sur l'identification des demandes, consultez Utilisation des journaux d’accès au serveur Amazon S3 pour identifier des demandes et Identification des demandes Amazon S3 à l'aide CloudTrail.

Vérifiez et mettez à jour les stratégies de compartiment qui utilisent des clés de condition associées à ACL

Une fois que vous avez appliqué le paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL, les nouveaux objets peuvent être téléchargés dans votre compartiment uniquement si la demande utilise des listes ACL de contrôle total du propriétaire du compartiment ou ne spécifie pas de liste ACL. Avant de désactiver les listes ACL, consultez votre stratégie de compartiment pour les clés de condition associées à la liste ACL.

Si votre stratégie de compartiment utilise une clé de condition associée à la liste ACL pour exiger la liste ACL `bucket-owner-full-control` prédéfinie (par exemple, `s3:x-amz-acl`), vous n'avez pas besoin de mettre à jour votre stratégie de compartiment. La stratégie de compartiment suivante utilise la `s3:x-amz-acl` pour exiger la liste ACL `bucket-owner-full-control` prédéfinie pour les demandes `PutObject` S3. Cette politique demande encore au rédacteur d'objets qu'il spécifie la liste ACL `bucket-owner-full-control` prédéfinie. Toutefois, les compartiments dont les listes ACL sont désactivées acceptent toujours cette liste ACL, de sorte que les demandes continuent d'aboutir sans modification côté client requise.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Cependant, si votre stratégie de compartiment utilise une clé de condition associée à la liste ACL qui nécessite une liste ACL différente, vous devez supprimer cette clé de condition. Cet exemple de stratégie de compartiment nécessite la liste ACL `public-read` pour les demandes S3 `PutObject` et doit donc être mis à jour avant de désactiver les listes ACL.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with public read access",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
            "Condition": {
```

```
                "StringEquals": {
                    "s3:x-amz-acl": "public-read"
                }
            }
        }
    ]
}
```

Autorisations de propriété d'objet

Pour appliquer, mettre à jour ou supprimer un paramètre de propriété d'objet pour un compartiment, vous devez utiliser l'autorisation `s3:PutBucketOwnershipControls`. Pour renvoyer le paramètre propriété de l'objet d'un compartiment, vous devez utiliser l'autorisation `s3:GetBucketOwnershipControls`. Pour plus d'informations, consultez Définition de la propriété d'objet lors de la création d'un compartiment et Affichage du paramètre de propriété d'objet pour un compartiment S3.

Désactivation des listes ACL pour tous les nouveaux compartiments

Par défaut, tous les nouveaux compartiments sont créés en appliquant le paramètre Propriétaire du compartiment appliqué, et les listes ACL sont désactivées. Nous vous recommandons de maintenir les listes ACL désactivées. En règle générale, nous recommandons d'utiliser des politiques basées sur les ressources S3 (politiques de compartiment et politiques de point d'accès) ou des politiques IAM pour le contrôle d'accès au lieu des ACL. Les politiques constituent une option de contrôle d'accès simplifiée et plus flexible. Les politiques de compartiment et de point d'accès vous permettent de définir des règles s'appliquant de manière générale à toutes les demandes adressées à vos ressources Amazon S3.

Réplication et propriété d'objet

Lorsque vous utilisez la réplication S3 et que les compartiments source et de destination appartiennent à des propriétaires différents Comptes AWS, vous pouvez désactiver les ACL (le paramètre propriétaire du compartiment étant imposé pour la propriété de l'objet) afin de remplacer la propriété de la réplique par Compte AWS celle du propriétaire du compartiment de destination. Ce paramètre imite le comportement de remplacement du propriétaire existant sans avoir besoin d'une autorisation `s3:ObjectOwnerOverrideToBucketOwner`. Tous les objets répliqués dans le compartiment de destination avec le paramètre Propriétaire du compartiment appliqué appartiennent au propriétaire du compartiment de destination. Pour plus d'informations sur l'option de remplacement

du propriétaire pour les configurations de réplication, consultez Modification du propriétaire d'un réplica.

Définition de la propriété de l'objet

Vous pouvez appliquer un paramètre de propriété d'objet à l'aide de la console Amazon S3 AWS CLI, AWS des SDK, de l'API REST Amazon S3 ou AWS CloudFormation. L'API et les AWS CLI commandes REST suivantes prennent en charge la propriété des objets :

API REST	AWS CLI	Description
PutBucketOwnershipControls	put-bucket-ownership-controls	Crée ou modifie le paramètre de propriété d'objet pour un compartiment S3 existant.
CreateBucket	create-bucket	Crée un compartiment à l'aide de l'en-tête de demande `x-amz-object-ownership` pour spécifier le paramètre de Propriété d'objet.
GetBucketOwnershipControls	get-bucket-ownership-controls	Extrait le paramètre de propriété d'objet pour un compartiment Amazon S3.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Supprime le paramètre de propriété d'objet pour un compartiment Amazon S3.

Pour plus d'informations sur l'application et l'utilisation des paramètres de propriété d'objet, consultez les rubriques suivantes.

Rubriques

- Conditions préalables à la désactivation des listes ACL
- Définition de la propriété d'objet lors de la création d'un compartiment
- Définition de la propriété d'un objet sur un compartiment existant
- Affichage du paramètre de propriété d'objet pour un compartiment S3

- Désactivation des listes ACL pour tous les nouveaux compartiments et application de la propriété des objets
- Résolution des problèmes

Conditions préalables à la désactivation des listes ACL

Si l'ACL de votre bucket accorde un accès en dehors de la Compte AWS vôtre, avant de désactiver les ACL, vous devez migrer les autorisations ACL de votre bucket vers votre politique de bucket et réinitialiser votre ACL de bucket sur l'ACL privée par défaut. Si vous ne migrez pas ces compartiments ACL, votre demande d'application du paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL échoue et renvoie le code d'erreur InvalidBucketAclWithObjectOwnership. Nous vous recommandons également de revoir les autorisations ACL de votre objet et de les migrer vers votre stratégie de compartiment. Pour obtenir plus d'informations sur les autres conditions préalables suggérées, consultez la page Conditions préalables à la désactivation des listes ACL.

Chacune de vos listes ACL de compartiment et d'objet existantes possède un équivalent dans une stratégie IAM. Les exemples de stratégie de compartiment suivants montrent comment les autorisations `READ` et `WRITE` pour les listes ACL de compartiment et d'objet correspondent aux autorisations IAM. Pour plus d'informations sur la façon dont chaque liste ACL se traduit par des autorisations IAM, veuillez consulter Mappage des autorisations de liste ACL et de stratégie d'accès.

Pour consulter et migrer les autorisations ACL vers des stratégies de compartiment, consultez les rubriques suivantes.

Rubriques

- Exemples de stratégies de compartiment
- Utilisation de la console S3 pour réviser et migrer les autorisations ACL
- Utilisation du AWS CLI pour vérifier et migrer les autorisations ACL
- Exemples de démonstrations

Exemples de stratégies de compartiment

Ces exemples de stratégies de compartiment vous montrent comment effectuer la migration des autorisations de `READ` et de `WRITE` pour les compartiments et les objets pour un Compte AWS tiers vers une stratégie de compartiment. Les listes ACL `READ_ACP` et `WRITE_ACP` sont moins pertinentes

pour les stratégies, car elles accordent des autorisations liées à l'ACL (`s3:GetBucketAcl`, `s3:GetObjectAcl`, `s3:PutBucketAcl`, et `s3:PutObjectAcl`).

Example : liste ACL **READ** pour un compartiment

Si votre bucket dispose d'une READ ACL qui Compte AWS *111122223333* autorise la liste du contenu de votre bucket, vous pouvez écrire une politique de bucket qui accorde `s3:ListBucket` `s3:ListBucketMultipartUploads` des autorisations pour votre bucket. `s3:ListBucketVersions`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to list the objects in a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [

                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        }
    ]
}
```

Example : listes ACL **READ** pour chaque objet d'un compartiment

Si chaque objet de votre compartiment dispose d'une READ ACL qui autorise l'accès à Compte AWS *111122223333*, vous pouvez rédiger une politique de compartiment qui accorde `s3:GetObject` des `s3:GetObjectVersion` autorisations à ce compte pour chaque objet de votre compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "Read permission for every object in a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

Cet exemple d'élément de ressource accorde l'accès à un objet spécifique.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Example : liste ACL **`WRITE`** qui accorde les autorisations d'écriture d'objets dans un compartiment

Si votre compartiment possède une `WRITE` ACL qui Compte AWS *111122223333* autorise l'écriture d'objets dans votre compartiment, vous pouvez rédiger une politique de compartiment qui accorde `s3:PutObject` l'autorisation pour votre compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to write objects to a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
```

```
    ]
}
```

Utilisation de la console S3 pour réviser et migrer les autorisations ACL

Pour passer en revue les autorisations ACL d'un compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, sélectionnez le nom de votre compartiment.
3. Choisissez l’onglet Permissions (Autorisations).
4. Sous Access control list (ACL) (Liste de contrôle d'accès [ACL]), vérifiez les autorisations ACL de votre compartiment.

Pour passer en revue les autorisations ACL d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient votre objet.
3. Dans la liste Objets, choisissez le nom de votre objet.
4. Choisissez l’onglet Permissions (Autorisations).
5. Sous Access control list (ACL) (Liste de contrôle d'accès [ACL]), vérifiez les autorisations ACL de votre objet.

Pour migrer les autorisations ACL et mettre à jour la liste ACL de votre compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, sélectionnez le nom de votre compartiment.
3. Dans l'onglet Permissions (Autorisations), sous Bucket Policy (Stratégie de compartiment), choisissez Edit (Modifier).
4. Dans la boîte Policy (Stratégie), ajoutez ou mettez à jour la stratégie de compartiment.

   Si vous souhaitez consulter des exemples de politiques de compartiment, consultez Exemples de stratégies de compartiment et Exemples de démonstrations.

5. Sélectionnez Enregistrer les modifications.
6. Mettez à jour la liste ACL de votre compartiment pour supprimer les autorisations ACL à d'autres groupes ou Comptes AWS.
7. Appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets.

Utilisation du AWS CLI pour vérifier et migrer les autorisations ACL

1. Pour renvoyer l'ACL du bucket correspondant à votre bucket, utilisez la get-bucket-acl AWS CLI commande suivante :

```
aws s3api get-bucket-acl --bucket DOC-EXAMPLE-BUCKET
```

Par exemple, cette liste ACL de compartiment accorde l'accès en `WRITE` et en `READ` à un compte tiers. Dans cette liste ACL, le compte tiers est identifié par l'ID d'utilisateur canonique. Pour appliquer le paramètre Propriétaire du compartiment appliqué et désactiver les listes ACL, vous devez migrer ces autorisations pour le compte tiers vers une politique de compartiment.

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                "ID":
 "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "READ"
```

```
        },
        {
            "Grantee": {
                "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                "ID":
 "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "WRITE"
        }
    ]
}
```

Pour d'autres exemples de listes ACL, voir Exemples de démonstrations.

2. Migrez les autorisations ACL de votre compartiment vers une stratégie de compartiment :

   Cet exemple de stratégie de compartiment accorde des autorisations `s3:PutObject` et `s3:ListBucket` pour un compte tiers. Dans la politique du compartiment, le compte tiers est identifié par l' Compte AWS ID (*111122223333*).

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PolicyForCrossAccountAllowUpload",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
             "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
             "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
```

```
    ]
}
```

Si vous souhaitez consulter d'autres exemples de politiques de compartiment, consultez Exemples de stratégies de compartiment et Exemples de démonstrations.

3. Pour renvoyer l'ACL pour un objet spécifique, utilisez la get-object-acl AWS CLI commande.

```
aws s3api get-object-acl --bucket DOC-EXAMPLE-BUCKET --key EXAMPLE-OBJECT-KEY
```

4. Si nécessaire, migrez les autorisations ACL d'objet vers votre stratégie de compartiment.

   Cet exemple d'élément de ressource accorde l'accès à un objet spécifique dans une stratégie de compartiment.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-OBJECT-KEY"
```

5. Réinitialisez la liste ACL de votre compartiment à la liste ACL par défaut.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

6. Appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets

Exemples de démonstrations

Les exemples suivants vous montrent comment migrer les autorisations ACL vers des stratégies de compartiment pour des cas d'utilisation spécifiques.

Rubriques

- Octroi de l’accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur
- Accordez un accès public en lecture aux objets se trouvant dans un compartiment.
- Accordez ElastiCache à Amazon pour Redis l'accès à votre compartiment S3

Octroi de l’accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur

Si vous souhaitez appliquer le paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL pour un compartiment de destination de journalisation des accès au serveur (également appelé compartiment cible), vous devez migrer les autorisations ACL de compartiment

pour le groupe de livraison des journaux S3 vers le principal du service de journalisation (`logging.s3.amazonaws.com`) dans une politique de compartiment. Pour plus d'informations sur les autorisations de la diffusion des journaux, consultez Autorisations de diffusion de journaux.

Ce compartiment ACL accorde un accès en `WRITE` et en `READ_ACP` au groupe de mise à disposition des journaux S3 :

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "Type": "CanonicalUser",
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "WRITE"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "READ_ACP"
        }
    ]
}
```

Pour migrer les autorisations ACL du compartiment pour le groupe de mise à disposition du journal S3 vers le principal du service de journalisation dans une stratégie de compartiment

1. Ajoutez la politique de compartiment suivante à votre compartiment de destination, en remplaçant les exemples de valeurs.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:      {
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/EXAMPLE-LOGGING-PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

2. Réinitialisez la liste ACL de votre compartiment de destination sur la liste ACL par défaut.

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. Appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets à votre compartiment de destination.

Accordez un accès public en lecture aux objets se trouvant dans un compartiment.

Si vos listes ACL d'objet accordent un accès public en lecture à tous les objets de votre compartiment, vous pouvez migrer ces autorisations ACL vers une stratégie de compartiment.

Cette liste ACL d'objet accorde un accès public en lecture à un objet dans un compartiment :

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

Pour migrer les autorisations ACL publiques en lecture vers une stratégie de compartiment

1. Pour accorder un accès public en lecture à tous les objets de votre compartiment, ajoutez la stratégie de compartiment suivante, en remplaçant les exemples de valeurs.

   ```
   aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

   policy.json:
   {
       "Version": "2012-10-17",
       "Statement": [
   ```

```
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

Pour accorder l'accès public à un objet spécifique dans une stratégie de compartiment, utilisez le format suivant pour l'élément `Resource`.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/OBJECT-KEY"
```

Pour accorder l'accès public à tous les objets avec un préfixe spécifique, utilisez le format suivant pour l'élément `Resource`.

```
"Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/PREFIX/*"
```

2. Appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets

Accordez ElastiCache à Amazon pour Redis l'accès à votre compartiment S3

Vous pouvez exporter votre sauvegarde ElastiCache pour Redis vers un compartiment S3, ce qui vous permet d'accéder à la sauvegarde depuis l'extérieur ElastiCache. Pour exporter votre sauvegarde vers un compartiment S3, vous devez autoriser ElastiCache la copie d'un instantané dans le compartiment. Si vous avez accordé des autorisations ElastiCache à une ACL de compartiment, vous devez migrer ces autorisations vers une politique de compartiment avant d'appliquer le paramètre imposé par le propriétaire du compartiment pour désactiver les ACL. Pour plus d'informations, consultez la section Accorder l' ElastiCache accès à votre compartiment Amazon S3 dans le guide de ElastiCache l'utilisateur Amazon.

L'exemple suivant montre les autorisations ACL du bucket qui accordent des autorisations à ElastiCache.

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "READ"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "WRITE"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "READ_ACP"
        }
    ]
```

```
}
```

Pour migrer les autorisations ACL du bucket ElastiCache pour Redis vers une politique de bucket

1. Ajoutez la stratégie de compartiment suivante à votre compartiment, en remplaçant les exemples de valeurs.

```
aws s3api put-bucket-policy --bucket DOC-EXAMPLE-BUCKET --policy file://policy.json

policy.json:
"Id": "Policy15397346",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "Region.elasticache-snapshot.amazonaws.com"
            },
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

2. Réinitialisez la liste ACL de votre compartiment à la liste ACL par défaut :

```
aws s3api put-bucket-acl --bucket DOC-EXAMPLE-BUCKET --acl private
```

3. Appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets

Définition de la propriété d'objet lors de la création d'un compartiment

Lorsque vous créez un compartiment, vous pouvez configurer la propriété d'objet S3. Pour définir la propriété d'un objet pour un compartiment existant, voir Définition de la propriété d'un objet sur un compartiment existant.

S3 Object Ownership est un paramètre Amazon S3 au niveau du compartiment que vous pouvez utiliser pour désactiver les listes de contrôle d'accès (ACL) et prendre possession de chaque objet de votre compartiment. Cela a pour effet de simplifier la gestion des accès aux données stockées dans Amazon S3. Par défaut, la propriété d'objets S3 est définie sur le paramètre Propriétaire du compartiment appliqué et les listes ACL sont désactivées pour les nouveaux compartiments. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient chaque objet présent dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet.

Object Ownership (Propriété de l'objet) dispose de trois paramètres que vous pouvez utiliser pour contrôler la propriété des objets téléchargés dans votre compartiment pour désactiver ou activer les listes ACL :

Listes ACL désactivées

- Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations sur les données du compartiment S3. Le compartiment utilise des stratégies pour définir le contrôle des accès.

Listes ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.
- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

Permissions (Autorisations) : pour appliquer le paramètre Bucket owner enforced (Propriétaire du compartiment imposé) ou le paramètre Bucket owner preferred (Propriétaire du compartiment préféré), vous devez disposer des autorisations suivantes : `s3:CreateBucket` et

`s3:PutBucketOwnershipControls`. Aucune autorisation supplémentaire n'est nécessaire lors de la création d'un compartiment avec le paramètre Object writer (Rédacteur d'objets) appliqué. Pour plus d'informations sur les autorisations Amazon S3, consultez Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

> **Important**
>
> La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL ; nous vous recommandons de désactiver les listes ACL, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès de chaque objet individuellement. Avec Object Ownership, vous pouvez désactiver les listes ACL et vous fier aux stratégies pour le contrôle des accès. Lorsque vous désactivez les ACL, vous pouvez facilement gérer un bucket contenant des objets chargés par différents AWS comptes. En tant que propriétaire du compartiment, vous êtes propriétaire de tous les objets du compartiment et pouvez gérer l'accès à ces derniers au moyen de stratégies.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer un bucket.

   > **Note**
   >
   > Pour limiter la latence et les coûts, et répondre aux exigences légales, choisissez une région proche de vous. Les objets stockés dans une Région ne la quittent jamais, sauf si vous les transférez explicitement vers une autre Région. Pour obtenir la liste d'Amazon S3 Régions AWS, consultez la section sur les Service AWS points de terminaison dans le Référence générale d'Amazon Web Services.

3. Dans le panneau de navigation de gauche, choisissez Compartiments.
4. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
5. Sous Configuration générale, consultez l' Région AWS endroit où votre bucket sera créé.

6. Sous Type de compartiment, sélectionnez Usage général.
7. Pour Nom du compartiment, saisissez le nom de votre compartiment.

   Le nom du compartiment doit présenter les caractéristiques suivantes :

   - Être unique dans une partition. Une partition est un regroupement de Régions. AWS dispose actuellement de trois partitions : `aws` (Régions Standard), `aws-cn` (Régions Chine) et `aws-us-gov` (AWS GovCloud (US) Regions).
   - Il doit comporter entre 3 et 63 caractères.
   - Être uniquement composé de lettres minuscules, de chiffres, de points (.) et de traits d'union (-). Pour une meilleure compatibilité, nous vous recommandons d'éviter d'utiliser des points (.) dans les noms de compartiment, à l'exception des compartiments utilisés uniquement pour l'hébergement de sites web statiques.
   - Commencer et se terminer par une lettre ou un chiffre.

   Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour plus d'informations sur l'attribution de noms à des compartiments, consultez Règles de dénomination de compartiment.

   > **Important**
   >
   > Évitez d'inclure des informations sensibles, notamment des numéros de compte, dans le nom du compartiment. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

8. AWS Management Console vous permet de copier les paramètres d'un bucket existant dans votre nouveau bucket. Si vous ne souhaitez pas copier les paramètres d'un bucket existant, passez à l'étape suivante.

   > **Note**
   >
   > Cette option :
   >
   > - N'est pas disponible dans le AWS CLI et n'est disponible que dans la console
   > - Non disponible pour les compartiments de répertoire
   > - Ne copie pas la politique du bucket du bucket existant vers le nouveau bucket

Pour copier les paramètres d'un compartiment existant, sous Copier les paramètres d'un compartiment existant, sélectionnez Choisir un compartiment. La fenêtre Choose bucket s'ouvre. Recherchez le compartiment contenant les paramètres que vous souhaitez copier, puis sélectionnez Choisir un compartiment. La fenêtre Choisir un compartiment se ferme et la fenêtre Créer un compartiment s'ouvre à nouveau.

Sous Copier les paramètres d'un bucket existant, vous pouvez maintenant voir le nom du bucket que vous avez sélectionné. Vous verrez également une option Restaurer les paramètres par défaut que vous pouvez utiliser pour supprimer les paramètres du bucket copiés. Passez en revue les autres paramètres du compartiment sur la page Créer un compartiment. Vous verrez qu'ils correspondent désormais aux paramètres du bucket que vous avez sélectionné. Vous pouvez passer à la dernière étape.

9. Sous Object Ownership (Propriété de l'objet), pour désactiver ou activer les listes ACL et contrôler la propriété des objets téléchargés dans votre compartiment, sélectionnez l'un des paramètres suivants :

   Listes ACL désactivées

   - Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations d'accès aux données du compartiment S3. Le compartiment utilise des stratégies exclusivement pour définir le contrôle des accès.

     Par défaut, les listes ACL sont désactivées. La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Pour plus d’informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

   Listes ACL activées

   - Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le

compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.

Si vous appliquez le paramètre Propriétaire du compartiment préféré, pour exiger que tous les chargements Amazon S3 incluent la liste ACL prédéfinie `bucket-owner-full-control`, vous pouvez ajouter une politique de compartiment qui autorise uniquement les chargements d'objets utilisant cette liste ACL.

- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

> **Note**
>
> Le paramètre par défaut est Propriétaire du compartiment appliqué. Pour appliquer le paramètre par défaut et maintenir les listes ACL désactivées, seule l'autorisation `s3:CreateBucket` est requise. Pour activer les listes ACL, vous devez disposez de l'autorisation `s3:PutBucketOwnershipControls`.

10. Dans Paramètres de blocage de l'accès public pour ce compartiment, choisissez les paramètres Bloquer l'accès public que vous souhaitez appliquer au compartiment.

    Par défaut, les quatre paramètres de blocage de l'accès public sont activés. Nous vous recommandons de maintenir tous les paramètres activés, sauf si vous savez que vous devez en désactiver un ou plusieurs pour votre cas d'utilisation spécifique. Pour en savoir plus sur le blocage de l'accès public, consultez Blocage de l'accès public à votre stockage Amazon S3.

    > **Note**
    >
    > Pour activer tous les paramètres de blocage de l'accès public, seule l'autorisation `s3:CreateBucket` est requise. Pour désactiver les paramètres de blocage de l'accès public, vous devez disposer de l'autorisation `s3:PutBucketPublicAccessBlock`.

11. (Facultatif) Sous Bucket Versioning (Gestion des versions du compartiment), vous pouvez choisir de conserver les variantes des objets dans votre compartiment. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3.

    Pour désactiver ou activer la gestion des versions sur votre compartiment, choisissez Disable (Désactiver) ou Enable (Activer).

12. (Facultatif) Sous Tags (Balises), vous pouvez choisir d'ajouter des balises à votre compartiment. Les balises sont des paires clé-valeur utilisées pour catégoriser le stockage.

    Pour ajouter une balise de compartiment, saisissez une Key (Clé) et éventuellement une Value (Valeur), puis choisissez Add Tag (Ajouter une balise).

13. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).

14. Pour configurer le chiffrement par défaut, dans Type de chiffrement, choisissez l'une des options suivantes :

    - Clés gérées par Amazon S3 (SSE-S3)
    - AWS Key Management Service clé (SSE-KMS)

    > **Important**
    >
    > Si vous utilisez l'option SSE-KMS pour votre configuration de chiffrement par défaut, vous êtes soumis aux quotas RPS (demandes par seconde) de AWS KMS. Pour plus d'informations sur les AWS KMS quotas et sur la manière de demander une augmentation de quota, consultez la section Quotas dans le guide du AWS Key Management Service développeur.

    Les compartiments et les nouveaux objets sont chiffrés à l'aide d'un chiffrement côté serveur avec une clé gérée par Amazon S3 comme niveau de base de configuration du chiffrement. Pour plus d'informations sur le chiffrement par défaut, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

    Pour en savoir plus sur l'utilisation du chiffrement côté serveur Amazon S3 pour chiffrer vos données, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

15. Si vous avez sélectionné CléAWS Key Management Service (SSE-KMS), procédez comme suit :

    a. Sous CléAWS KMS , spécifiez votre clé KMS de l'une des manières suivantes :

       - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis choisissez votre clé KMS dans la liste des clés disponibles.

         La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez

Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .

- Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
- Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

  Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

> **Important**
>
> Vous ne pouvez utiliser que les clés KMS disponibles dans le même compartiment Région AWS que le bucket. La console Amazon S3 répertorie uniquement les 100 premières clés KMS dans la même région que le compartiment. Pour utiliser une clé KMS qui n'est pas répertoriée, vous devez saisir l'ARN de votre clé KMS. Si vous souhaitez utiliser une clé KMS qui appartient à un autre compte, vous devez d'abord avoir l'autorisation d'utiliser cette clé KMS, puis saisir l'ARN de la clé KMS. Pour plus d'informations sur les autorisations entre comptes pour les clés KMS, consultez la section Creating KMS keys that other accounts can use (Création de clés KMS que d'autres comptes peuvent utiliser) dans le Guide du développeur AWS Key Management Service . Pour en savoir plus sur SSE-KMS, consultez Spécification du chiffrement côté serveur avec AWS KMS (SSE-KMS).
> Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 prend uniquement en charge les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur l'utilisation AWS KMS avec Amazon S3, consultezUtilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

b. Lorsque vous configurez votre compartiment pour utiliser le chiffrement par défaut avec SSE-KMS, vous pouvez également activer les clés de compartiment S3. Les clés de compartiment S3 réduisent le coût du chiffrement en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.

   Pour utiliser les clés de compartiment S3, sous la Clé de compartiment, choisissez Activer.

16. (Facultatif) Si vous souhaitez activer le verrouillage des objets S3, procédez comme suit :

    a. Choisissez Advanced Settings (Paramètres avancés).

    > **Important**
    >
    > L'activation du verrouillage d'objet active également la gestion des versions pour le compartiment. Après l'avoir activé, vous devez configurer les paramètres de conservation et de mise en suspens juridique par défaut du verrouillage d'objets pour protéger les nouveaux objets contre la suppression ou l'écrasement.

    b. Pour activer le verrouillage d'objets, choisissez Enable (Activer), lisez l'avertissement qui s'affiche et confirmez-le.

    Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

    > **Note**
    >
    > Pour créer un compartiment prenant en charge le verrouillage d'objets, vous devez disposer des autorisations suivantes : `s3:CreateBucket`, `s3:PutBucketVersioning` et `s3:PutBucketObjectLockConfiguration`.

17. Choisissez Créer un compartiment.

En utilisant le AWS CLI

Pour définir la propriété de l'objet lorsque vous créez un nouveau compartiment, utilisez la `create-bucket` AWS CLI commande avec le `--object-ownership` paramètre.

Cet exemple applique le paramètre Propriétaire du compartiment appliqué à un nouveau compartiment à l'aide de l'interface AWS CLI :

```
aws s3api create-bucket --bucket  DOC-EXAMPLE-BUCKET --region us-east-1 --object-
ownership BucketOwnerEnforced
```

> **Important**
>
> Si vous ne définissez pas la propriété de l'objet lorsque vous créez un bucket à l'aide de AWS CLI, le paramètre par défaut sera `ObjectWriter` (ACL activées).

Utilisation du AWS SDK pour Java

Cet exemple définit le paramètre Propriétaire du compartiment appliqué à un nouveau compartiment à l'aide du kit AWS SDK for Java :

```
    // Build the ObjectOwnership for CreateBucket
    CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
            .build()

     // Send the request to Amazon S3
     s3client.createBucket(createBucketRequest);
```

En utilisant AWS CloudFormation

Pour utiliser la `AWS::S3::Bucket` AWS CloudFormation ressource afin de définir la propriété de l'objet lorsque vous créez un nouveau bucket, reportez-vous OwnershipControlsAWS::S3::Bucketau Guide de l'AWS CloudFormation utilisateur.

Utilisation de l'API REST

Pour appliquer le paramètre Propriétaire du compartiment appliqué pour la propriété d'objets S3, utilisez l'opération d'API `CreateBucket` avec l'en-tête de demande `x-amz-object-ownership` défini sur `BucketOwnerEnforced`. Pour plus d'informations et des exemples, consultez CreateBucket dans la Référence d'API Amazon Simple Storage Service.

Étapes suivantes : une fois que vous avez appliqué les paramètres Propriétaire du compartiment appliqué ou Propriétaire du compartiment préféré pour Propriété d’objets, vous pouvez suivre les étapes suivantes :

- Bucket owner enforced (Appliqué par le propriétaire du compartiment) – Vous pouvez exiger que tous les nouveaux compartiments soient créés avec des listes ACL désactivées à l'aide d'une politique IAM ou Organizations.
- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Ajoutez une stratégie de compartiment S3 pour exiger que la liste ACL `bucket-owner-full-control` prédéfinie pour tous les téléchargements d'objets vers votre compartiment.

Définition de la propriété d'un objet sur un compartiment existant

Vous pouvez configurer la propriété des objets S3 sur un compartiment S3 existant. Pour appliquer la propriété d'objet lors de la création d'un compartiment, consultez Définition de la propriété d'objet lors de la création d'un compartiment.

S3 Object Ownership est un paramètre Amazon S3 au niveau du compartiment que vous pouvez utiliser pour désactiver les listes de contrôle d'accès (ACL) et prendre possession de chaque objet de votre compartiment. Cela a pour effet de simplifier la gestion des accès aux données stockées dans Amazon S3. Par défaut, la propriété d'objets S3 est définie sur le paramètre Propriétaire du compartiment appliqué et les listes ACL sont désactivées pour les nouveaux compartiments. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient chaque objet présent dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet.

Object Ownership (Propriété de l'objet) dispose de trois paramètres que vous pouvez utiliser pour contrôler la propriété des objets téléchargés dans votre compartiment pour désactiver ou activer les listes ACL :

Listes ACL désactivées

- Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations sur les données du compartiment S3. Le compartiment utilise des stratégies pour définir le contrôle des accès.

Listes ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.
- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

Prérequis : avant d'appliquer le paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL, vous devez migrer les autorisations ACL de compartiment vers des politiques de compartiment et réinitialiser vos listes ACL de compartiment vers la liste ACL privée par défaut. Nous vous recommandons également de migrer les autorisations ACL d'objet vers des stratégies de compartiment et de modifier les stratégies de compartiment qui nécessitent des listes ACL autres que les listes ACL de contrôle total du propriétaire du compartiment. Pour plus d'informations, consultez Conditions préalables à la désactivation des listes ACL.

Autorisations : pour pouvoir utiliser cette opération, vous devez disposer de l'autorisation `s3:PutBucketOwnershipControls`. Pour plus d'informations sur les autorisations Amazon S3, consultez Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment auquel vous souhaitez appliquer le paramètre de propriété d'objet S3.
3. Choisissez l'onglet Permissions (Autorisations).
4. Sous Object Ownership (Propriétaire de l'objet), sélectionnez Edit (Modifier).
5. Sous Object Ownership (Propriété de l'objet), pour désactiver ou activer les listes ACL et contrôler la propriété des objets téléchargés dans votre compartiment, sélectionnez l'un des paramètres suivants :

   Listes ACL désactivées

   - Bucket owner enforced (Appliqué par le propriétaire du compartiment) – Les listes ACL sont désactivées, et le propriétaire du compartiment possède automatiquement tous les objets du

compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations sur les données du compartiment S3. Le compartiment utilise des stratégies pour définir le contrôle des accès.

Pour exiger que tous les nouveaux compartiments soient créés avec les ACL désactivées à l'aide d'IAM ou de AWS Organizations politiques, consultez. Désactivation des listes ACL pour tous les nouveaux compartiments (application du propriétaire du compartiment)

Listes ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.

  Si vous appliquez le paramètre préféré par le propriétaire du compartiment pour exiger que tous les téléchargements Amazon S3 incluent la liste ACL `bucket-owner-full-control`prête à l'emploi, vous pouvez ajouter une stratégie de compartiment qui autorise uniquement les téléchargements d'objets utilisant cette liste ACL.
- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

6. Choisissez Enregistrer.

En utilisant le AWS CLI

Pour appliquer un paramètre de propriété d'objet à un compartiment existant, utilisez la commande `put-bucket-ownership-controls`avec le paramètre `--ownership-controls`. Les valeurs valides pour la propriété sont `BucketOwnerEnforced`, `BucketOwnerPreferred` ou `ObjectWriter`.

Cet exemple applique le paramètre Propriétaire du compartiment appliqué pour un compartiment existant à l'aide de l' AWS CLI :

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Pour plus d'informations sur `put-bucket-ownership-controls`, veuillez consulter la rubrique put-bucket-ownership-controls dans le Guide de l'utilisateur AWS Command Line Interface .

Utilisation du AWS SDK pour Java

Cet exemple applique le paramètre `BucketOwnerEnforced` de Propriété d'objet à un compartiment existant à l'aide d' AWS SDK for Java :

```
        // Build the ObjectOwnership for BucketOwnerEnforced
        OwnershipControlsRule rule = OwnershipControlsRule.builder()
               .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
               .build();

        OwnershipControls ownershipControls = OwnershipControls.builder()
                  .rules(rule)
                  .build()

         // Build the PutBucketOwnershipControlsRequest
         PutBucketOwnershipControlsRequest putBucketOwnershipControlsRequest =
               PutBucketOwnershipControlsRequest.builder()
                       .bucket(BUCKET_NAME)
                       .ownershipControls(ownershipControls)
                       .build();

         // Send the request to Amazon S3
         s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

En utilisant AWS CloudFormation

À utiliser AWS CloudFormation pour appliquer un paramètre de propriété d'objet à un bucket existant, consultez AWS::S3::Bucket OwnershipControlsle guide de AWS CloudFormation l'utilisateur.

Utilisation de l'API REST

Pour utiliser l'API REST afin d'appliquer un paramètre de propriété d'objet à un compartiment S3 existant, utilisez `PutBucketOwnershipControls`. Pour plus d'informations, veuillez consulter PutBucketOwnershipControls dans la Référence d'API Amazon Simple Storage Service.

Étapes suivantes : une fois que vous avez appliqué les paramètres Propriétaire du compartiment appliqué ou Propriétaire du compartiment préféré pour Propriété d’objets, vous pouvez suivre les étapes suivantes :

- Bucket owner enforced (Appliqué par le propriétaire du compartiment) – Vous pouvez exiger que tous les nouveaux compartiments soient créés avec des listes ACL désactivées à l'aide d'une politique IAM ou Organizations.

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Ajoutez une stratégie de compartiment S3 pour exiger que la liste ACL `bucket-owner-full-control` prédéfinie pour tous les téléchargements d'objets vers votre compartiment.

Affichage du paramètre de propriété d'objet pour un compartiment S3

S3 Object Ownership est un paramètre Amazon S3 au niveau du compartiment que vous pouvez utiliser pour désactiver les listes de contrôle d'accès (ACL) et prendre possession de chaque objet de votre compartiment. Cela a pour effet de simplifier la gestion des accès aux données stockées dans Amazon S3. Par défaut, la propriété d'objets S3 est définie sur le paramètre Propriétaire du compartiment appliqué et les listes ACL sont désactivées pour les nouveaux compartiments. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient chaque objet présent dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet.

Object Ownership (Propriété de l'objet) dispose de trois paramètres que vous pouvez utiliser pour contrôler la propriété des objets téléchargés dans votre compartiment pour désactiver ou activer les listes ACL :

Listes ACL désactivées

- Propriétaire du compartiment appliqué (par défaut) : les listes ACL sont désactivées et le propriétaire du compartiment détient automatiquement chaque objet présent dans le compartiment et en a le contrôle total. Les listes ACL n'affectent plus les autorisations sur les données du compartiment S3. Le compartiment utilise des stratégies pour définir le contrôle des accès.

Listes ACL activées

- Bucket owner preferred (Préféré par le propriétaire du compartiment) – Le propriétaire du compartiment possède les nouveaux objets que d'autres comptes écrivent dans le compartiment avec la liste ACL `bucket-owner-full-control` prête à l'emploi, et en a le contrôle total.
- Auteur d'objets : celui Compte AWS qui télécharge un objet est propriétaire de l'objet, en a le contrôle total et peut autoriser d'autres utilisateurs à y accéder via des ACL.

Vous pouvez afficher les paramètres de propriété d'objet S3 pour un compartiment Amazon S3 Pour définir la propriété d'un objet pour un nouveau compartiment, voir Définition de la propriété d'objet

lors de la création d'un compartiment. Pour définir la propriété d'un objet pour un compartiment existant, voir Définition de la propriété d'un objet sur un compartiment existant.

Autorisations : pour pouvoir utiliser cette opération, vous devez disposer de l'autorisation `s3:GetBucketOwnershipControls`. Pour plus d'informations sur les autorisations Amazon S3, consultez Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment auquel vous souhaitez appliquer le paramètre de propriété d'objet.
3. Choisissez l'onglet Permissions (Autorisations).
4. Sous Object Ownership (Propriétaire de l'objet), vous pouvez voir les paramètres de propriété des objets pour votre compartiment.

En utilisant le AWS CLI

Pour récupérer le paramètre de propriété de l'objet S3 pour un compartiment S3, utilisez la get-bucket-ownership-controls AWS CLI commande.

```
aws s3api get-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET
```

Utilisation de l'API REST

Pour récupérer le paramètre Propriété de l'objet pour un compartiment S3, utilisez l'opération `GetBucketOwnershipControls` d'API. Pour plus d'informations, consultez GetBucketOwnershipControls.

Désactivation des listes ACL pour tous les nouveaux compartiments et application de la propriété des objets

Nous vous recommandons de désactiver les listes ACL sur vos compartiments Amazon S3. Vous pouvez le faire en appliquant le paramètre Propriétaire du compartiment appliqué pour la propriété d'objets S3. Lorsque vous appliquez ce paramètre, les listes ACL sont désactivées, vous possédez automatiquement tous les objets de votre compartiment et vous en avez le contrôle total. Pour exiger

que tous les nouveaux compartiments soient créés avec les ACL désactivées, utilisez les politiques AWS Identity and Access Management (IAM) ou les politiques de contrôle des AWS Organizations services (SCP), comme décrit dans la section suivante.

Pour imposer la propriété des objets aux nouveaux objets sans désactiver les listes ACL, vous pouvez appliquer le paramètre préféré du propriétaire du compartiment. Lorsque vous appliquez ce paramètre, nous vous recommandons vivement de mettre à jour votre politique de compartiment afin d'exiger la liste ACL `bucket-owner-full-control` prédéfinie pour toutes les demandes PUT envoyées à votre compartiment. Les clients doivent également être mis à jour pour envoyer la liste ACL `bucket-owner-full-control` prédéfinie dans votre compartiment à partir d'autres comptes.

Rubriques

- Désactivation des listes ACL pour tous les nouveaux compartiments (application du propriétaire du compartiment)
- Exiger l' bucket-owner-full-controlACL prédéfini pour les PUT opérations Amazon S3 (préféré du propriétaire du compartiment)

Désactivation des listes ACL pour tous les nouveaux compartiments (application du propriétaire du compartiment)

L'exemple suivant de politique IAM refuse l'autorisation `s3:CreateBucket` pour un rôle ou un utilisateur IAM spécifique, sauf si le paramètre Propriétaire du compartiment appliqué est appliqué pour Propriété d'objets. La paire de clé-valeur dans le bloc `Condition` spécifie `s3:x-amz-object-ownership` comme sa clé et le paramètre `BucketOwnerEnforced` comme sa valeur. En d'autres termes, l'utilisateur IAM peut créer des compartiments uniquement s'il définit le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets et désactive les listes ACL. Vous pouvez également utiliser cette politique comme SCP limite pour votre AWS organisation.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireBucketOwnerFullControl",
            "Action": "s3:CreateBucket",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"
```

```
                }
            }
        }
    ]
}
```

Exiger l' bucket-owner-full-controlACL prédéfini pour les **PUT** opérations Amazon S3 (préféré du propriétaire du compartiment)

Avec le paramètre préféré du propriétaire du compartiment pour la propriété de l'objet, vous, en tant que propriétaire du compartiment, possédez et avez le contrôle total sur les nouveaux objets que d'autres comptes écrivent dans votre compartiment avec la liste ACL `bucket-owner-full-control` prédéfinie. Toutefois, si d'autres comptes écrivent des objets dans votre compartiment sans la liste ACL `bucket-owner-full-control` prédéfinie, le rédacteur d'objets conserve son accès total. En tant que propriétaire du compartiment, vous pouvez implémenter une stratégie de compartiment qui n'autorise les écritures que si elles spécifient la liste ACL `bucket-owner-full-control` prédéfinie.

> **Note**
>
> Si les listes ACL sont désactivées avec le paramètre Propriétaire du compartiment appliqué, en tant que propriétaire du compartiment, vous possédez automatiquement tous les objets de votre compartiment et en avez le contrôle total. Vous n'avez pas besoin d'utiliser cette section pour mettre à jour votre stratégie de compartiment afin d'imposer la propriété de l'objet au propriétaire du compartiment.

La politique de compartiment suivante spécifie que le compte *111122223333* ne peut charger des objets *DOC-EXAMPLE-BUCKET* que si la liste ACL de l'objet est définie sur `bucket-owner-full-control`. Veillez à remplacer *111122223333* par voter compte et *DOC-EXAMPLE-BUCKET* par le nom de votre compartiment.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Only allow writes to my bucket with bucket owner full control",
         "Effect": "Allow",
         "Principal": {
            "AWS": [
```

```
                "arn:aws:iam::111122223333:user/ExampleUser"
            ]
         },
         "Action": [
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
         "Condition": {
            "StringEquals": {
               "s3:x-amz-acl": "bucket-owner-full-control"
            }
         }
      }
   ]
}
```

Voici un exemple d'opération de copie qui inclut la liste ACL `bucket-owner-full-control` prédéfinie à l'aide d' AWS Command Line Interface (AWS CLI).

```
aws s3 cp file.txt s3://DOC-EXAMPLE-BUCKET --acl bucket-owner-full-control
```

Une fois la politique de compartiment mise en vigueur, si le client n'inclut pas la liste ACL prédéfinie `bucket-owner-full-control`, l'opération échoue et le chargeur reçoit l'erreur suivante :

Une erreur s'est produite (AccessDenied) lors de l'appel de l' PutObject opération : Accès refusé.

> **Note**
>
> Si les clients doivent accéder aux objets après le chargement, vous devez accorder des autorisations supplémentaires au compte de chargement. Pour en savoir plus sur l'octroi aux comptes de l'accès à vos ressources, consultez Procédures pas à pas utilisant des politiques pour gérer l'accès à vos ressources Amazon S3.

Résolution des problèmes

Lorsque vous appliquez le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets S3, les listes de contrôle d'accès (ACL) sont désactivées et vous, en tant que propriétaire du compartiment, possédez automatiquement tous les objets dans votre compartiment. Les listes ACL n'affectent plus les autorisations sur les objets dans votre compartiment. Vous pouvez utiliser

des politiques pour accorder des autorisations. Toutes les demandes S3 PUT doivent spécifier la liste ACL prédéfinie `bucket-owner-full-control` ou ne pas spécifier de liste ACL, sinon ces demandes échouent. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Si une liste ACL non valide est spécifiée ou si les autorisations ACL de compartiment accordent un accès hors de votre Compte AWS, les erreurs suivantes peuvent s'afficher.

AccessControlListNotSupported

Une fois que vous avez appliqué le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, les listes ACL sont désactivées. Les demandes de définition ou de mise à jour des listes de contrôle d'accès échouent avec une `400` erreur et renvoient le code AccessControlListNotSupportedd'erreur. Les demandes de lecture de listes ACL sont toujours prises en charge. Les demandes de lecture des listes ACL renvoient toujours une réponse qui affiche un contrôle total pour le propriétaire du compartiment. Dans vos opérations PUT, vous devez spécifier des listes ACL de contrôle total du propriétaire du compartiment ou ne pas spécifier de liste ACL. Dans le cas contraire, vos opérations PUT échouent.

L'exemple de `put-object` AWS CLI commande suivant inclut l'`public-read`ACL prédéfinie.

```
aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key object-key-name --body doc-example-body --acl public-read
```

Si le compartiment utilise le paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL, cette opération échoue et le chargeur reçoit le message d'erreur suivant :

Une erreur s'est produite (AccessControlListNotSupported) lors de l'appel de l'PutObject opération : le bucket n'autorise pas les ACL

InvalidBucketAclWithObjectOwnership

Si vous souhaitez appliquer le paramètre Propriétaire du compartiment appliqué pour désactiver les listes ACL, votre liste ACL de compartiment doit donner le contrôle total uniquement au propriétaire du compartiment. L'ACL de votre bucket ne peut pas donner accès à un groupe externe Compte AWS ou à un autre groupe. Par exemple, si votre `CreateBucket` demande impose le nom du propriétaire du bucket et spécifie une ACL du bucket qui donne accès à une Compte AWS adresse externe, votre demande échoue avec une `400` erreur et renvoie le code InvalidBucketAclWithObjectOwnershipd'erreur. De même, si votre demande `PutBucketOwnershipControls` définit le paramètre appliqué par le propriétaire du compartiment

sur un compartiment doté d'une liste ACL de compartiment qui accorde des autorisations à d'autres utilisateurs, la demande échoue.

Example : Une liste ACL de compartiment existante accorde un accès public en lecture.

Par exemple, si une liste ACL de compartiment existante accorde un accès public en lecture, vous ne pouvez pas appliquer le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets tant que vous n'avez pas migré ces autorisations ACL vers une politique de compartiment et que vous n'avez pas réinitialisé votre liste ACL de compartiment à la liste ACL privée par défaut. Pour plus d'informations, consultez Conditions préalables à la désactivation des listes ACL.

Cet exemple de liste ACL de compartiment existante accorde un accès public en lecture.

```
{
    "Owner": {
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

L'exemple de `put-bucket-ownership-controls` AWS CLI commande suivant applique le paramètre imposé par le propriétaire du bucket pour Object Ownership :

```
aws s3api put-bucket-ownership-controls --bucket DOC-EXAMPLE-BUCKET --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Étant donné que la liste ACL du compartiment accorde un accès public en lecture, la demande échoue et renvoie le code d'erreur suivant :

Une erreur s'est produite (InvalidBucketAclWithObjectOwnership) lors de l'appel de l' PutBucketOwnershipControls opération : les ACL du bucket ne peuvent pas être définies avec ObjectOwnership le paramètre BucketOwnerEnforced

Utilisation du partage des ressources entre origines multiples (CORS)

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine. Le CORS vous permet de créer de riches applications web côté client avec Amazon S3 et d'autoriser de manière sélective un accès cross-origin à vos ressources Amazon S3.

Cette section fournit une présentation du CORS. Les sous-rubriques décrivent comment activer CORS à l'aide de la console Amazon S3 ou par programmation à l'aide de l'API REST Amazon S3 et des kits SDK. AWS

Partage des ressources de plusieurs origines : scénarios de cas d'utilisation

Les exemples de scénarios suivants utilisent le partage CORS.

Scénario 1

Supposons que vous hébergiez un site Internet dans un compartiment Amazon S3 appelé `website`, comme décrit dans Hébergement d'un site Web statique à l'aide d'Amazon S3. Les utilisateurs chargent le point de terminaison du site web :

```
http://website.s3-website.us-east-1.amazonaws.com
```

Vous souhaitez maintenant utiliser les pages Web stockées dans ce compartiment pour pouvoir effectuer des requêtes GET et PUT authentifiées JavaScript sur le même compartiment en utilisant le point de terminaison de l'API Amazon S3 pour le compartiment,. `website.s3.us-east-1.amazonaws.com` Un navigateur bloque normalement l'autorisation JavaScript de ces demandes, mais avec CORS, vous pouvez configurer votre compartiment pour activer explicitement les demandes provenant de différentes origines. `website.s3-website.us-east-1.amazonaws.com`

Scénario 2

Supposons que vous souhaitez héberger une police web à partir de votre compartiment S3. Une fois encore, les navigateurs requièrent un contrôle CORS (aussi appelé contrôle en amont) pour le chargement des polices web. Vous configurez le compartiment qui héberge la police web pour permettre à toute origine d'effectuer ces demandes.

Comment Amazon S3 évalue la configuration CORS sur un compartiment ?

Lorsque Amazon S3 reçoit une demande en amont d'un navigateur, il évalue la configuration CORS du compartiment et utilise la première règle `CORSRule` qui correspond à la demande entrante du navigateur pour permettre une demande cross-origin. Pour qu'une règle corresponde, les conditions suivantes doivent être remplies :

- L'en-tête `Origin` de la demande doit correspondre à un élément `AllowedOrigin`.
- La méthode de demande (par exemple, GET ou PUT) ou l'en-tête `Access-Control-Request-Method` dans le cas d'une demande `OPTIONS` en amont doit être l'un des éléments `AllowedMethod`.
- Chaque en-tête listé dans l'en-tête `Access-Control-Request-Headers` de la demande sur la demande en amont doit correspondre à un élément `AllowedHeader`.

> **Note**
>
> Les listes ACL et stratégies continuent de s'appliquer lorsque vous activer le CORS sur le compartiment.

Comment le point d'accès Object Lambda prend en charge le CORS

Quand S3 Object Lambda reçoit une demande d'un navigateur ou que la demande inclut un en-tête `Origin`, S3 Object Lambda ajoute toujours un champ d'en-tête `“AllowedOrigins":"*"`.

Pour plus d'informations sur l'utilisation du CORS, consultez les rubriques suivantes.

Rubriques

- Configuration CORS
- Configuration du partage des ressources entre origines multiples (CORS)

Configuration CORS

Pour configurer le compartiment afin qu'il autorise les demandes cross-origin, vous créez une configuration CORS. La configuration CORS est un document contenant les règles identifiant les origines auxquelles vous autorisez l'accès au compartiment, les opérations (méthodes HTTP) prises en charge pour chaque origine, et d'autres informations propres aux opérations. Vous pouvez ajouter jusqu'à 100 règles à la configuration. Vous pouvez ajouter la configuration CORS en tant que sous-ressource `cors` au compartiment.

Si vous configurez le CORS dans la console S3, vous devez utiliser JSON pour créer une configuration CORS. La nouvelle console S3 ne prend en charge que les configurations JSON CORS.

Pour plus d'informations sur la configuration CORS et les éléments qu'elle contient, consultez les rubriques ci-dessous. Pour savoir comment ajouter une configuration CORS, consultez la section Configuration du partage des ressources entre origines multiples (CORS).

> **Important**
>
> Dans la console S3, la configuration CORS doit être de type JSON.

Rubriques

Exemple 1

Au lieu d'accéder à un site web en utilisant un point de terminaison de site web Amazon S3, vous pouvez utiliser votre propre domaine, comme `example1.com`, pour proposer votre contenu. Pour

plus d'informations sur l'utilisation de votre propre domaine, consultez Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

L'exemple de configuration `cors` suivant possède trois règles, qui sont spécifiées comme éléments `CORSRule` :

- La première règle autorise des demandes cross-origin PUT, POST et DELETE de l'origine `http://www.example1.com`. La règle autorise également tous les en-têtes dans une demande OPTIONS en amont via l'en-tête `Access-Control-Request-Headers`. En réponse à toute demande OPTIONS en amont, Amazon S3 renvoie les en-têtes demandés.
- La deuxième règle autorise les mêmes demandes cross-origin que la première, mais elle s'applique à une autre origine, `http://www.example2.com`.
- La troisième règle permet des demandes GET cross-origin de toutes les origines. Le caractère générique * fait référence à toutes les origines.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example1.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
```

```
        "AllowedOrigins": [
            "http://www.example2.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example1.com</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>

   <AllowedHeader>*</AllowedHeader>
 </CORSRule>
 <CORSRule>
   <AllowedOrigin>http://www.example2.com</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>

   <AllowedHeader>*</AllowedHeader>
 </CORSRule>
 <CORSRule>
   <AllowedOrigin>*</AllowedOrigin>
   <AllowedMethod>GET</AllowedMethod>
 </CORSRule>
</CORSConfiguration>
```

Exemple 2

La configuration CORS permet également d'ajouter des paramètres facultatifs, comme illustré dans la configuration CORS suivante. Dans cet exemple, la configuration CORS permet les demandes PUT, POST et DELETE cross-origin depuis l'origine `http://www.example.com`.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example.com"
        ],
        "ExposeHeaders": [
            "x-amz-server-side-encryption",
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example.com</AllowedOrigin>
   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>
   <AllowedHeader>*</AllowedHeader>
  <MaxAgeSeconds>3000</MaxAgeSeconds>
  <ExposeHeader>x-amz-server-side-encryption</
ExposeHeader>
  <ExposeHeader>x-amz-request-id</
```

```
ExposeHeader>
  <ExposeHeader>x-amz-id-2</ExposeHeader>
 </CORSRule>
</CORSConfiguration>
```

L'élément `CORSRule` dans la configuration précédente inclut les éléments facultatifs suivants :

- `MaxAgeSeconds` : spécifie le nombre d'heures en secondes (dans cet exemple, 3 000) pendant lesquelles le navigateur met en cache une réponse Amazon S3 à une demande OPTIONS en amont pour la ressource spécifiée. Le fait de mettre en cache la réponse évite au navigateur de devoir envoyer les demandes en amont à Amazon S3 si la demande originale est répétée.
- `ExposeHeader`—Identifie les en-têtes de réponse (dans cet exemple`x-amz-server-side-encryption`,`x-amz-request-id`, et`x-amz-id-2`) auxquels les clients peuvent accéder depuis leurs applications (par exemple, depuis un JavaScript `XMLHttpRequest` objet).

AllowedMethod élément

Dans la configuration CORS, vous pouvez spécifier les valeurs suivantes pour l'élément `AllowedMethod`.

- GET
- PUT
- POST
- DELETE
- HEAD

AllowedOrigin élément

Dans l'élément `AllowedOrigin`, vous spécifiez les origines à partir desquelles vous souhaitez autoriser des demandes cross-domaine, par exemple, `http://www.example.com`. La chaîne de caractères d'origine peut contenir au maximum un caractère générique `*`, comme `http://*.example.com`. Si vous le souhaitez, vous pouvez spécifier `*` comme l'origine pour permettre à toutes les origines d'envoyer des demandes cross-origin. Vous pouvez également spécifier `https` pour permettre uniquement les origines sécurisées.

AllowedHeader élément

L'élément `AllowedHeader` spécifie les en-têtes autorisés dans une demande en amont via l'en-tête `Access-Control-Request-Headers`. Chaque nom d'en-tête dans l'en-tête `Access-Control-Request-Headers` doit correspondre à une entrée dans la règle. Amazon S3 envoie uniquement dans une réponse les en-têtes autorisés qui ont été demandés. Pour obtenir un exemple de liste d'en-têtes pouvant être utilisés dans les demandes adressées à Amazon S3, veuillez accéder à En-têtes de demande courants dans la Référence d'API Amazon Simple Storage Service.

Chaque AllowedHeader chaîne de la règle peut contenir au maximum un caractère générique*. Par exemple, `<AllowedHeader>x-amz-*</AllowedHeader>` permet tous les en-têtes propres à Amazon.

ExposeHeader élément

Chaque `ExposeHeader` élément identifie un en-tête dans la réponse auquel vous souhaitez que les clients puissent accéder depuis leurs applications (par exemple, depuis un JavaScript `XMLHttpRequest` objet). Pour obtenir la liste des en-têtes de réponse Amazon S3 courants, veuillez accéder à En-têtes de réponse courants dans la Référence d'API Amazon Simple Storage.

MaxAgeSeconds élément

L'élément `MaxAgeSeconds` spécifie le nombre d'heures en secondes pendant lesquelles le navigateur peut mettre en cache une réponse pour une demande en amont comme identifié par la ressource, la méthode HTTP et l'origine.

Configuration du partage des ressources entre origines multiples (CORS)

Le partage des ressources cross-origin (CORS) définit un moyen pour les applications Web clientes chargées dans un domaine particulier d'interagir avec les ressources d'un autre domaine. Le CORS vous permet de créer de riches applications web côté client avec Amazon S3 et d'autoriser de manière sélective un accès cross-origin à vos ressources Amazon S3.

Cette section explique comment activer CORS à l'aide de la console Amazon S3, de l'API REST Amazon S3 et des AWS kits de développement logiciel. Pour configurer le compartiment afin d'autoriser les demandes entre origines multiples, vous devez ajouter une configuration CORS au compartiment. Une configuration CORS est un document qui définit les règles identifiant les origines auxquelles vous autorisez l'accès au compartiment, les opérations (méthodes HTTP) prises en charge pour chaque origine, et d'autres informations propres aux opérations. Dans la console S3, la configuration CORS doit être un document JSON.

Pour obtenir un exemple de configurations CORS en JSON et XML, veuillez consulter Configuration CORS.

Utilisation de la console S3

Cette section explique comment utiliser la console Amazon S3 pour ajouter une configuration CORS (partage des ressources cross-origin) à un compartiment S3.

Lorsque vous activez CORS sur le compartiment, les listes de contrôle d'accès (ACL) et les autres stratégies d'autorisation d'accès continuent à s'appliquer.

> **Important**
>
> Dans la nouvelle console S3, la configuration CORS doit être de type JSON. Pour obtenir des exemples de configurations CORS en JSON et XML, consultez la section Configuration CORS.

Pour ajouter une configuration CORS à un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez créer une stratégie de compartiment.
3. Choisissez Permissions.
4. Dans la section Cross-origin resource sharing (CORS) (Partage des ressources cross-origin (CORS)), choisissez Edit (Modifier).
5. Dans la zone de texte Editeur de configuration CORS, tapez ou copiez et collez une nouvelle configuration CORS, ou modifiez une configuration existante.

   La configuration CORS est un fichier JSON. Le texte que vous saisissez dans l'éditeur doit être dans un format JSON valide. Pour plus d'informations, consultez Configuration CORS.
6. Sélectionnez Save Changes (Enregistrer les modifications).

> **Note**
>
> Amazon S3 affiche l'Amazon Resource Name (ARN) du compartiment en regard du titre Editeur de configuration CORS. Pour plus d'informations sur les ARN, consultez

les sections Amazon Resource Names (ARN) et AWS Service Namespaces dans le. Référence générale d'Amazon Web Services

Utilisation des AWS SDK

Vous pouvez utiliser le AWS SDK pour gérer le partage de ressources entre origines (CORS) pour un bucket. Pour plus d'informations sur le CORS, consultez Utilisation du partage des ressources entre origines multiples (CORS).

Les exemples suivants :

- Crée une configuration CORS et définit la configuration sur un compartiment
- Récupère une configuration et la modifie en ajoutant une règle
- Ajoute la configuration modifiée au compartiment
- Supprime la configuration

Java

Example

Example

Pour obtenir des instructions sur la création et le test d'un échantillon de travail, reportez-vous à la section Getting Started du Guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketCrossOriginConfiguration;
import com.amazonaws.services.s3.model.CORSRule;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
```

```
public class CORS {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        // Create two CORS rules.
        List<CORSRule.AllowedMethods> rule1AM = new
 ArrayList<CORSRule.AllowedMethods>();
        rule1AM.add(CORSRule.AllowedMethods.PUT);
        rule1AM.add(CORSRule.AllowedMethods.POST);
        rule1AM.add(CORSRule.AllowedMethods.DELETE);
        CORSRule rule1 = new
 CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
                .withAllowedOrigins(Arrays.asList("http://*.example.com"));

        List<CORSRule.AllowedMethods> rule2AM = new
 ArrayList<CORSRule.AllowedMethods>();
        rule2AM.add(CORSRule.AllowedMethods.GET);
        CORSRule rule2 = new
 CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
                .withAllowedOrigins(Arrays.asList("*")).withMaxAgeSeconds(3000)
                .withExposedHeaders(Arrays.asList("x-amz-server-side-encryption"));

        List<CORSRule> rules = new ArrayList<CORSRule>();
        rules.add(rule1);
        rules.add(rule2);

        // Add the rules to a new CORS configuration.
        BucketCrossOriginConfiguration configuration = new
 BucketCrossOriginConfiguration();
        configuration.setRules(rules);

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Add the configuration to the bucket.
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Retrieve and display the configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
```

```
            printCORSConfiguration(configuration);

            // Add another new rule.
            List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
            rule3AM.add(CORSRule.AllowedMethods.HEAD);
            CORSRule rule3 = new
CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
                    .withAllowedOrigins(Arrays.asList("http://www.example.com"));

            rules = configuration.getRules();
            rules.add(rule3);
            configuration.setRules(rules);
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Verify that the new rule was added by checking the number of rules in
the
            // configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            System.out.println("Expected # of rules = 3, found " +
configuration.getRules().size());

            // Delete the configuration.
            s3Client.deleteBucketCrossOriginConfiguration(bucketName);
            System.out.println("Removed CORS configuration.");

            // Retrieve and display the configuration to verify that it was
            // successfully deleted.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            printCORSConfiguration(configuration);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void printCORSConfiguration(BucketCrossOriginConfiguration
configuration) {
        if (configuration == null) {
```

```
            System.out.println("Configuration is null.");
        } else {
            System.out.println("Configuration has " +
 configuration.getRules().size() + " rules\n");

            for (CORSRule rule : configuration.getRules()) {
                System.out.println("Rule ID: " + rule.getId());
                System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
                System.out.println("AllowedMethod: " + rule.getAllowedMethods());
                System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
                System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
                System.out.println("ExposeHeader: " + rule.getExposedHeaders());
                System.out.println();
            }
        }
    }
}
```

.NET

Example

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CORSTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

```
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
        {
            try
            {
                // Create a new configuration request and add two rules
                CORSConfiguration configuration = new CORSConfiguration
                {
                    Rules = new System.Collections.Generic.List<CORSRule>
                        {
                          new CORSRule
                          {
                            Id = "CORSRule1",
                            AllowedMethods = new List<string> {"PUT", "POST",
 "DELETE"},
                            AllowedOrigins = new List<string> {"http://
*.example.com"}
                          },
                          new CORSRule
                          {
                            Id = "CORSRule2",
                            AllowedMethods = new List<string> {"GET"},
                            AllowedOrigins = new List<string> {"*"},
                            MaxAgeSeconds = 3000,
                            ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
                          }
                        }
                };

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Retrieve an existing configuration.
                configuration = await RetrieveCORSConfigurationAsync();

                // Add a new rule.
                configuration.Rules.Add(new CORSRule
                {
                    Id = "CORSRule3",
```

```
                    AllowedMethods = new List<string> { "HEAD" },
                    AllowedOrigins = new List<string> { "http://www.example.com" }
                });

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Verify that there are now three rules.
                configuration = await RetrieveCORSConfigurationAsync();
                Console.WriteLine();
                Console.WriteLine("Expected # of rulest=3; found:{0}",
configuration.Rules.Count);
                Console.WriteLine();
                Console.WriteLine("Pause before configuration delete. To continue,
click Enter...");
                Console.ReadKey();

                // Delete the configuration.
                await DeleteCORSConfigurationAsync();

                // Retrieve a nonexistent configuration.
                configuration = await RetrieveCORSConfigurationAsync();
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
        {

            PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };
```

```
            var response = await s3Client.PutCORSConfigurationAsync(request);
        }

        static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
        {
            GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
            {
                BucketName = bucketName

            };
            var response = await s3Client.GetCORSConfigurationAsync(request);
            var configuration = response.Configuration;
            PrintCORSRules(configuration);
            return configuration;
        }

        static async Task DeleteCORSConfigurationAsync()
        {
            DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest
            {
                BucketName = bucketName
            };
            await s3Client.DeleteCORSConfigurationAsync(request);
        }

        static void PrintCORSRules(CORSConfiguration configuration)
        {
            Console.WriteLine();

            if (configuration == null)
            {
                Console.WriteLine("\nConfiguration is null");
                return;
            }

            Console.WriteLine("Configuration has {0} rules:",
configuration.Rules.Count);
            foreach (CORSRule rule in configuration.Rules)
            {
                Console.WriteLine("Rule ID: {0}", rule.Id);
                Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
                Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
rule.AllowedMethods.ToArray()));
```

```
                Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
 rule.AllowedOrigins.ToArray()));
                Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
 rule.AllowedHeaders.ToArray()));
                Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
 rule.ExposeHeaders.ToArray()));
            }
        }
    }
}
```

Utilisation de l'API REST

Pour définir une configuration CORS sur votre compartiment, vous pouvez utiliser AWS Management Console. Si l'application l'exige, vous pouvez également envoyer directement des demandes REST. Les sections suivantes de la Référence d'API Amazon Simple Storage Service décrivent les actions de l'API REST liées à la configuration CORS :

- PutBucketCors
- GetBucketCors
- DeleteBucketCors
- OPTIONS object

Journalisation et surveillance dans Amazon S3

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon S3 et de vos AWS solutions. Vous devez collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une défaillance multipoint, le cas échéant. AWS fournit plusieurs outils pour surveiller vos ressources Amazon S3 et répondre aux incidents potentiels.

Pour plus d'informations, consultez Surveillance d'Amazon S3.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

CloudWatch Alarmes Amazon

À l'aide des CloudWatch alarmes Amazon, vous observez une seule métrique sur une période que vous spécifiez. Si la métrique dépasse un seuil donné, une notification est envoyée à une rubrique ou AWS Auto Scaling à une politique Amazon SNS. CloudWatch les alarmes n'appellent pas d'actions car elles se trouvent dans un état particulier. L'état doit avoir changé et avoir été conservé pendant un nombre de périodes spécifié. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.

AWS CloudTrail Journaux

CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans Amazon S3. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon S3, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires. Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.

Amazon GuardDuty

Amazon GuardDuty est un service de détection des menaces qui surveille en permanence vos comptes, vos conteneurs, vos charges de travail et les données de votre AWS environnement afin d'identifier les menaces ou les risques de sécurité potentiels pour vos compartiments

S3. GuardDuty fournit également un contexte détaillé sur les menaces détectées. GuardDuty surveille les journaux AWS CloudTrail de gestion pour détecter les menaces et affiche les informations relatives à la sécurité. Par exemple, GuardDuty inclura les facteurs d'une demande d'API, tels que l'utilisateur qui a fait la demande, le lieu d'où la demande a été faite et l'API spécifique demandée, qui peuvent être inhabituels dans votre environnement. GuardDuty S3 Protection surveille les événements de données S3 collectés par CloudTrail et identifie les comportements potentiellement anormaux et malveillants dans tous les compartiments S3 de votre environnement.

Journaux d'accès Amazon S3

Les journaux d'accès au serveur fournissent des enregistrements détaillés sur les demandes formulées à un compartiment. Les journaux d'accès au serveur sont utiles pour de nombreuses applications. Par exemple, les informations des journaux d'accès peuvent s'avérer utiles en cas d'audit de sécurité ou d'audit des accès. Pour plus d'informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.

AWS Trusted Advisor

Trusted Advisor s'appuie sur les meilleures pratiques apprises en servant des centaines de milliers de AWS clients. Trusted Advisor inspecte votre AWS environnement, puis émet des recommandations lorsque des opportunités se présentent pour économiser de l'argent, améliorer la disponibilité et les performances du système ou contribuer à combler les failles de sécurité. Tous les AWS clients ont accès à cinq Trusted Advisor chèques. Les clients disposant d'un plan de support Business ou Enterprise peuvent consulter tous les Trusted Advisor chèques.

Trusted Advisor comporte les vérifications suivantes relatives à Amazon S3 :

- Configuration de journalisation des compartiments Amazon S3.
- Vérifications de sécurité pour les compartiments Amazon S3 dont les autorisations permettent un libre accès.
- Vérifications de la tolérance aux pannes pour les compartiments Amazon S3 pour lesquels la gestion des versions est désactivée ou suspendue.

Pour plus d'informations, consultez AWS Trusted Advisor dans le Guide de l'utilisateur AWS Support .

Les bonnes pratiques de sécurité suivantes s'appliquent également à la consignation et à la surveillance :

- Identify and audit all your Amazon S3 buckets
- Implement monitoring using Amazon Web Services monitoring tools
- Activer AWS Config
- Enable Amazon S3 server access logging
- Use CloudTrail
- Monitor Amazon Web Services security advisories

Validation de conformité pour Amazon S3

La sécurité et la conformité d'Amazon S3 sont évaluées par des auditeurs tiers dans le cadre de plusieurs programmes de AWS conformité, notamment les suivants :

- System and Organization Controls (SOC)
- Norme de sécurité des données de l'industrie des cartes de paiement (PCI DSS)
- Federal Risk and Authorization Management Program (FedRAMP)
- Health Insurance Portability and Accountability Act (HIPAA)

AWS fournit une liste fréquemment mise à jour des AWS services concernés par des programmes de conformité spécifiques sur la page AWS Services in Scope by Compliance Program.

Les rapports d'audit tiers peuvent être téléchargés à l'aide de AWS Artifact. Pour plus d'informations, consultez la section Téléchargement de rapports dans AWS Artifact.

Pour plus d'informations sur les programmes de AWS conformité, consultez AWS la section Programmes de conformité.

Votre responsabilité de conformité lors de l'utilisation d'Amazon S3 est déterminée par la sensibilité de vos données, les objectifs de conformité de votre organisation, ainsi que par la législation et la réglementation applicables. Si votre utilisation d'Amazon S3 est soumise à la conformité à des normes telles que HIPAA, PCI, ou FedRAMP, AWS fournit des ressources pour vous aider :

- Guides de démarrage rapide sur la sécurité et la conformité qui abordent les considérations architecturales et les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- Architecting for HIPAA Security and Compliance décrit la manière dont les entreprises les aident AWS à répondre aux exigences de la HIPAA.
- AWS Les ressources de conformité fournissent plusieurs classeurs et guides différents qui peuvent s'appliquer à votre secteur d'activité et à votre région.
- AWS Config Le service peut permettre d'évaluer comment les configurations de vos ressources se conforment aux pratiques internes, aux normes et aux directives industrielles.
- AWS Security Hub vous fournit une vue complète de l'état de votre sécurité interne AWS et vous aide à vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.
- Utilisation du verrouillage des objets S3 peut vous aider à répondre aux exigences techniques de régulateurs des services financiers (comme les organismes SEC, FINRA et CFTC) qui exigent

un stockage des données en écriture seule et lectures multiples (WORM) pour certains types de document et informations d'enregistrement.

- Inventaire Simple Storage Service (Amazon S3) peut vous aider à auditer et signaler le statut de réplication et de chiffrement de vos objets à des fins professionnelles, de conformité et d'obligations réglementaires.

Résilience dans Amazon S3

L'infrastructure AWS mondiale est construite autour des régions et des zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées connectées par un réseau à faible latence, à haut débit et hautement redondant. Ces zones de disponibilité vous offrent un moyen efficace de concevoir et d'exploiter des applications et des bases de données. Elles sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données. Si vous avez spécifiquement besoin de répliquer vos données sur de plus grandes distances géographiques, vous pouvez utiliser Vue d'ensemble de la réplication d'objets ce qui permet de copier automatiquement et de manière asynchrone des objets entre des compartiments situés dans différents compartiments. Régions AWS

Chacune Région AWS possède plusieurs zones de disponibilité. Vous pouvez déployer vos applications dans plusieurs zones de disponibilité au sein d'une même Région pour bénéficier d'une tolérance aux pannes et d'une faible latence. Les zones de disponibilité sont connectées les unes aux autres via une mise en réseau à fibres optiques rapide et privée, ce qui nous permet de concevoir facilement des applications qui basculent automatiquement entre les zones de disponibilité sans interruption.

Pour plus d'informations sur les zones de disponibilité Régions AWS et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Outre l'infrastructure AWS mondiale, Amazon S3 propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Configuration du cycle de vie

Une configuration du cycle de vie est un ensemble de règles qui définit des action qu'Amazon S3 applique à un groupe d'objets. Avec des règles de configuration du cycle de vie, vous pouvez indiquer à Amazon S3 de passer à des classes de stockage moins onéreuses, de les archiver ou de les supprimer. Pour plus d’informations, consultez Gestion du cycle de vie de votre stockage.

Contrôle de version

La gestion des versions est un moyen de conserver plusieurs variantes d'un objet dans le même compartiment. Vous pouvez utiliser le contrôle de version pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans votre compartiment Amazon S3. Le contrôle de version permet de récupérer facilement les données en cas d'actions involontaires des utilisateurs ou de défaillances des applications. Pour plus d’informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Verrouillage des objets S3

La fonctionnalité de verrouillage des objets S3 vous permet de stocker des objets selon un modèle WORM write once, read many. En utilisant le verrouillage des objets S3, vous pouvez empêcher qu'un objet soit supprimé ou remplacé sur une période déterminée ou indéfinie. La fonctionnalité de verrouillage des objets S3 vous permet de satisfaire aux exigences réglementaires qui nécessitent le stockage WORM, ou de simplement ajouter une couche supplémentaire de protection contre la suppression et les modifications d'objet. Pour plus d’informations, consultez Utilisation du verrouillage des objets S3.

Classes de stockage

Amazon S3 offre une gamme de classes de stockage à choisir en fonction des exigences de votre charge de travail. Les classes de stockage S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent sont conçues pour les données auxquelles vous accédez environ une fois par mois et nécessitent un accès en millisecondes. La classe de stockage S3 Glacier Instant Retrieval est conçue pour les données d'archivage de longue durée accessibles avec un accès en millisecondes auxquelles vous accédez environ une fois par trimestre. Pour les données d'archivage qui ne nécessitent pas d'accès immédiat, telles que les sauvegardes, vous pouvez utiliser les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Pour plus d’informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Les bonnes pratiques de sécurité suivantes s'appliquent également à la résilience :

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Chiffrement des sauvegardes Amazon S3

Si vous stockez des sauvegardes à l'aide d'Amazon S3, le chiffrement de vos sauvegardes dépend de la configuration de ces compartiments. Amazon S3 fournit un moyen de définir le comportement de chiffrement par défaut pour un compartiment S3. Vous pouvez définir le chiffrement par défaut sur un compartiment afin que tous les objets soient chiffrés lorsqu'ils sont stockés dans le compartiment. Le chiffrement par défaut prend en charge les clés stockées dans AWS KMS (SSE-KMS). Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.

Pour en savoir plus sur le contrôle de version et le verrouillage des objets, consultez les rubriques suivantes : Utilisation de la gestion des versions dans les compartiments S3 Utilisation du verrouillage des objets S3

Sécurité de l'infrastructure dans Amazon S3

En tant que service géré, Amazon S3 est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le pilier de sécurité du AWS Well-Architected Framework.

L'accès à Amazon S3 via le réseau se fait via des API AWS publiées. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2. Nous recommandons également la prise en charge de TLS 1.3. (Pour plus d'informations sur cette recommandation, consultez la section Connexions AWS cloud plus rapides avec TLS 1.3 sur le blog AWS de sécurité.) Les clients doivent également prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). En outre, les demandes doivent être signées à l'aide de AWS la signature V4 ou de AWS la signature V2, ce qui nécessite la fourniture d'informations d'identification valides.

Ces API peuvent être appelées depuis n'importe quel emplacement réseau. Toutefois, Amazon S3 prend bel et bien en charge les stratégies d'accès basées sur les ressources, ce qui peut inclure des restrictions en fonction de l'adresse IP source. Vous pouvez aussi utiliser des stratégies de compartiment Amazon S3 pour contrôler l'accès aux compartiments à partir de points de terminaison de Virtual Private Cloud (VPC) spécifiques ou de VPC spécifiques. En fait, cela isole l'accès réseau à un compartiment Amazon S3 donné uniquement du VPC spécifique au sein AWS du réseau. Pour plus d’informations, consultez Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.

Les bonnes pratiques de sécurité suivantes s'appliquent également à la sécurité de l'infrastructure dans Amazon S3 :

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Configuration et analyse des vulnérabilités dans Amazon S3

AWS gère les tâches de sécurité de base telles que l'application de correctifs au système d'exploitation client (OS) et aux bases de données, la configuration du pare-feu et la reprise après sinistre. Ces procédures ont été vérifiées et certifiées par les tiers appropriés. Pour plus de détails, consultez les ressources suivantes :

- Validation de conformité pour Amazon S3
- Modèle de responsabilité partagée
- Amazon Web Services : Présentation des procédures de sécurité

Les bonnes pratiques de sécurité suivantes s'appliquent également à la configuration et à l'analyse des vulnérabilités dans Amazon S3 :

- Identify and audit all your Amazon S3 buckets
- Activer AWS Config

Bonnes pratiques de sécurité pour Amazon S3

Amazon S3 fournit différentes fonctions de sécurité à prendre en compte lorsque vous développez et implémentez vos propres stratégies de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des recommandations utiles plutôt que comme des prescriptions.

Rubriques

Bonnes pratiques en matière de sécurité Amazon S3

Les bonnes pratiques suivantes pour Amazon S3 peuvent vous aider à éviter des incidents de sécurité.

Désactiver les listes de contrôle d'accès (ACL)

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété d'objets est définie sur le paramètre Propriétaire du compartiment appliqué, et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets présents dans le compartiment et gère l'accès aux données exclusivement à l'aide de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation de listes de contrôle d'accès (ACL). Nous vous recommandons de désactiver les listes ACL, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès pour chaque objet individuellement. Pour désactiver les listes de contrôle d'accès et prendre possession de tous les objets de votre compartiment, appliquez le paramètre d'application du propriétaire du compartiment pour la propriété des objets S3. Lorsque vous désactivez les listes ACL, vous pouvez facilement gérer un compartiment avec des objets téléchargés par différents Comptes AWS.

Lorsque les listes ACL sont désactivées, le contrôle d'accès à vos données est basé sur des politiques telles que les suivantes :

- AWS Identity and Access Management politiques utilisateur (IAM)
- Politiques de compartiment S3
- Politiques de point de terminaison de cloud privé virtuel (VPC)
- AWS Organizations politiques de contrôle des services (SCP)

La désactivation des ACL simplifie la gestion des autorisations et l'audit. Les listes ACL sont désactivées par défaut pour les nouveaux compartiments. Vous pouvez également désactiver les listes ACL pour les compartiments existants. Si vous avez un compartiment contenant déjà des objets, une fois que vous avez désactivé les listes ACL, les listes ACL d'objet et de compartiment ne font plus partie du processus d'évaluation des accès. Au lieu de cela, l'accès est accordé ou refusé sur la base des politiques.

Avant de désactiver les listes ACL, assurez-vous d'effectuer les opérations suivantes :

- Passez en revue votre politique de compartiment pour vous assurer qu'elle couvre toutes les façons dont vous avez l'intention d'accorder l'accès à votre compartiment hors de votre compte.
- Réinitialisez votre liste ACL de compartiment à sa valeur par défaut (contrôle total accordé au propriétaire du compartiment).

Une fois que vous avez désactivé les listes ACL, les comportements suivants se produisent :

- Votre compartiment accepte uniquement les demandes PUT qui ne spécifient pas de liste ACL ou les demandes PUT avec des listes ACL de contrôle total du propriétaire du compartiment. Ces listes ACL incluent la liste ACL prédéfinie `bucket-owner-full-control` ou des formes équivalentes de cette liste ACL exprimées en XML.
- Les applications existantes prenant en charge les listes ACL de contrôle total du propriétaire du compartiment n'ont aucun impact.
- PUTles requêtes contenant d'autres ACL (par exemple, des autorisations personnalisées accordées à certaines Comptes AWS) échouent et renvoient un code d'état HTTP `400 (Bad Request)` avec le code `AccessControlListNotSupported` d'erreur.

Pour plus d’informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Veillez à ce que vos compartiments Amazon S3 utilisent les stratégies appropriées et ne soient pas accessibles publiquement

À moins que vous ayez explicitement besoin que quiconque sur Internet puisse lire ou écrire dans votre compartiment S3, veillez à ce que votre compartiment S3 ne soit pas public. Voici quelques-unes des étapes que vous pouvez suivre pour bloquer l'accès public :

- Utilisez le blocage de l'accès public S3. Le blocage de l'accès public S3 vous permet de configurer facilement des contrôles centralisés pour limiter l'accès public à vos ressources Amazon S3. Ces contrôles centralisés sont appliqués quelle que soit la manière dont les ressources sont créées. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.
- Identifiez les politiques de compartiment Amazon S3 qui autorisent une identité à caractère générique telle que `"Principal": "*"` (qui signifie en fait « n'importe qui »). Recherchez également les politiques qui autorisent une action à caractère générique `"*"` (qui permet à l'utilisateur d'effectuer n'importe quelle action dans le compartiment Amazon S3).
- De même, recherchez les listes de contrôle d'accès aux compartiments (ACL) Amazon S3 qui fournissent un accès en lecture, en écriture ou complet à « tout le monde » ou à « tout utilisateur authentifié AWS ».
- Utilisez l'opération d'API `ListBuckets` pour analyser tous vos compartiments Amazon S3. Utilisez ensuite `GetBucketAcl`, `GetBucketWebsite` et `GetBucketPolicy` pour déterminer si chaque compartiment dispose de contrôles d'accès et d'une configuration conformes.
- Utilisez AWS Trusted Advisor pour inspecter votre implémentation Amazon S3.
- Envisagez de mettre en œuvre des contrôles de détection continus en utilisant les AWS Config Rules gérés s3-bucket-public-read-prohibited et s3-bucket-public-write-prohibited.

Pour plus d'informations, consultez Identity and Access Management pour Amazon S3.

Identifiez les menaces potentielles qui pèsent sur vos compartiments Amazon S3 à l'aide d'Amazon GuardDuty

Amazon GuardDuty est un service de détection des menaces qui identifie les menaces potentielles qui pèsent sur vos comptes, vos conteneurs, vos charges de travail et les données de votre AWS environnement. En utilisant des modèles d'apprentissage automatique (ML) et des fonctionnalités de détection des anomalies et des menaces, Amazon surveille GuardDuty en permanence différentes sources de données afin d'identifier et de hiérarchiser les risques de sécurité potentiels et les activités malveillantes dans votre environnement. Lorsque vous l'activez GuardDuty, il permet de détecter les menaces pour les sources de données de base, notamment

les événements AWS CloudTrail de gestion, les journaux de flux VPC et les journaux DNS. Pour étendre la détection des menaces aux événements du plan de données dans les compartiments S3, vous pouvez activer la fonction de protection GuardDuty S3. Cette fonctionnalité détecte les menaces telles que l'exfiltration de données et l'accès suspect aux compartiments S3 via des nœuds Tor. GuardDutyétablit également un modèle de référence normal dans votre environnement et, lorsqu'il identifie un comportement potentiellement anormal, il fournit des informations contextuelles pour vous aider à corriger le compartiment S3 ou les informations d'identification potentiellement compromis. AWS Pour plus d'informations, consultez GuardDuty.

Implémentation d'un accès sur la base du moindre privilège

Lorsque vous accordez des autorisations, vous décidez qui obtient quelles autorisations pour telles ou telles ressources Amazon S3. Vous activez des actions spécifiques que vous souhaitez autoriser sur ces ressources. Par conséquent, nous vous recommandons d'accorder uniquement les autorisations qui sont requises pour effectuer une tâche. L'implémentation d'un accès sur la base du moindre privilège est fondamentale pour réduire les risques en matière de sécurité et l'impact que pourraient avoir des d'erreurs ou des actes de malveillance.

Les outils suivants sont disponibles pour l'implémentation d'un accès sur la base du moindre privilège :

- Actions politiques pour Amazon S3 et Limites d'autorisations pour les entités IAM
- Comment Amazon S3 fonctionne avec IAM
- Présentation de la liste de contrôle d'accès (ACL)
- Politiques de contrôle de service

Pour obtenir des conseils sur ce qu'il faut prendre en compte lors du choix d'un ou plusieurs des mécanismes précédents, veuillez consulter Identity and Access Management pour Amazon S3.

Utiliser des rôles IAM pour les applications Services AWS qui nécessitent un accès à Amazon S3

Pour que les applications exécutées sur Amazon EC2 ou autre puissent accéder Services AWS aux ressources Amazon S3, elles doivent inclure des AWS informations d'identification valides dans leurs demandes d' AWS API. Nous vous recommandons de ne pas stocker AWS les informations d'identification directement dans l'application ou l'instance Amazon EC2. Il s'agit d'autorisations à long terme qui ne font pas automatiquement l'objet d'une rotation et qui pourraient avoir un impact commercial important si elles étaient compromises.

À la place, utilisez un rôle IAM pour gérer des autorisations temporaires pour les applications ou services devant accéder à Amazon S3. Lorsque vous utilisez un rôle, vous n'avez pas à distribuer

des informations d'identification à long terme (telles qu'un nom d'utilisateur et un mot de passe ou des clés d'accès) à une instance Amazon EC2 ou Service AWS, par exemple. AWS Lambda Le rôle fournit des autorisations temporaires que les applications peuvent utiliser lorsqu'elles appellent d'autres AWS ressources.

Pour plus d'informations, consultez les rubriques suivantes dans le Guide de l'utilisateur IAM :

- Rôles IAM
- Scénarios courants pour les rôles : utilisateurs, applications et services.

Prise en compte du chiffrement de données au repos

Pour protéger des données au repos dans Amazon S3, les options suivantes sont possibles :

- Chiffrement côté serveur : le chiffrement est configuré par défaut pour tous les compartiments Amazon S3, et tous les nouveaux objets chargés dans un compartiment S3 sont automatiquement chiffrés au repos. Le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) est la configuration de chiffrement par défaut pour chaque compartiment dans Amazon S3. Pour utiliser un autre type de chiffrement, vous pouvez soit spécifier le type de chiffrement côté serveur à utiliser dans vos demandes PUT S3, soit définir la configuration de chiffrement par défaut dans le compartiment de destination.

  Amazon S3 propose également les options de chiffrement côté serveur suivantes :

  - Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS)
  - Chiffrement double couche côté serveur avec clés AWS Key Management Service (AWS KMS) (DSSE-KMS)
  - Chiffrement côté serveur avec clés fournies par le client (SSE-C)

  Pour plus d'informations, consultez Protection des données avec le chiffrement côté serveur.
- Chiffrement côté client : chiffrez les données côté client et chargez les données chiffrées dans Amazon S3. Dans ce cas, vous gérez le processus de chiffrement, les clés de chiffrement et les outils associés. Comme le chiffrement côté serveur, le chiffrement côté client peut contribuer à réduire les risques en chiffrant les données avec une clé qui est stockée dans un autre mécanisme que le mécanisme qui stocke les données elles-mêmes.

  Amazon S3 fournit plusieurs options de chiffrement côté client. Pour plus d'informations, consultez Protection des données avec le chiffrement côté client.

Application du chiffrement des données en transit

Vous pouvez utiliser le protocole HTTPS (TLS) pour empêcher les attaquants potentiels d'espionner ou de manipuler le trafic réseau en utilisant des attaques similaires. person-in-the-middle Nous vous recommandons d'autoriser uniquement les connexions chiffrées sur HTTPS (TLS) en utilisant la condition aws:SecureTransport dans vos politiques de compartiment Amazon S3.

> **Important**
>
> Nous recommandons à votre application de ne pas épingler les certificats TLS Amazon S3 car AWS cela ne prend pas en charge l'épinglage de certificats approuvés par le public. S3 renouvelle automatiquement les certificats et le renouvellement peut avoir lieu à tout moment avant l'expiration des certificats. Le renouvellement d'un certificat génère une nouvelle paire de clés publique-privée. Si vous avez épinglé un certificat S3 récemment renouvelé avec une nouvelle clé publique, vous ne pourrez pas vous connecter à S3 tant que votre application n'aura pas utilisé le nouveau certificat.

Envisagez également de mettre en œuvre des contrôles de détection continus en utilisant la règle AWS Config gérée s3-bucket-ssl-requests-only.

Prise en compte de l'utilisation du verrouillage des objets S3

Avec le verrouillage des objets S3, vous pouvez stocker des objets en utilisant un modèle « Write Once Read Many » (WORM). Le verrouillage des objets S3 peut contribuer à empêcher une suppression accidentelle ou inappropriée de données. Par exemple, vous pouvez utiliser S3 Object Lock pour protéger vos AWS CloudTrail journaux.

Pour plus d’informations, consultez Utilisation du verrouillage des objets S3.

Activation de la gestion des versions S3

La gestion des versions S3 est un moyen de conserver plusieurs variantes d'un objet dans le même compartiment. Vous pouvez utiliser la gestion des versions pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans votre compartiment . La gestion des versions permet de récupérer facilement les données en cas d'actions involontaires des utilisateurs ou de défaillances des applications.

Envisagez également de mettre en œuvre des contrôles de détection continus en utilisant la règle AWS Config gérée s3-bucket-versioning-enabled.

Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Prise en compte de l'utilisation de la réplication entre régions Amazon S3

Même si Amazon S3 stocke par défaut vos données dans plusieurs zones de disponibilité distinctes géographiquement, les exigences de conformité peuvent vous obliger à stocker les données à des distances encore plus importantes. Avec S3 Cross-Region Replication (CRR), vous pouvez répliquer des données entre des sites distants Régions AWS pour répondre à ces exigences. Le CRR permet la copie automatique et asynchrone d'objets dans différents compartiments. Régions AWS Pour plus d'informations, consultez Vue d'ensemble de la réplication d'objets.

> **Note**
>
> La réplication CRR exige que la gestion des versions soit activée pour les compartiments S3 source et de destination.

Envisagez également de mettre en œuvre des contrôles de détection continus en utilisant la règle AWS Config gérée s3-bucket-replication-enabled.

Prise en compte de l'utilisation des points de terminaison de VPC pour l'accès Amazon S3

Un point de terminaison de VPC (Virtual Private Cloud) pour Amazon S3 est une entité logique au sein d'un VPC qui autorise la connectivité uniquement à Amazon S3. Les points de terminaison VPC peuvent empêcher le trafic de traverser le réseau Internet ouvert.

Les points de terminaison VPC pour Amazon S3 offrent plusieurs façons de contrôler l'accès à vos données Amazon S3 :

- Vous pouvez contrôler les demandes, les utilisateurs ou les groupes autorisés à traverser un point de terminaison de VPC spécifique en utilisant des politiques de compartiment S3.
- Vous pouvez contrôler quels VPC ou points de terminaison d'un VPC ont accès à vos compartiments S3 en utilisant des stratégies de compartiment S3.
- Vous pouvez éviter l'exfiltration de données en utilisant un VPC qui ne comporte pas de passerelle Internet.

Pour plus d'informations, consultez Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.

Utilisez des services AWS de sécurité gérés pour surveiller la sécurité des données

Plusieurs services AWS de sécurité gérés peuvent vous aider à identifier, évaluer et surveiller les risques de sécurité et de conformité pour vos données Amazon S3. Ces services peuvent également vous aider à protéger vos données contre ces risques. Ces services incluent des fonctionnalités de détection, de surveillance et de protection automatisées conçues pour passer des ressources Amazon S3 réservées à une seule personne Compte AWS à des ressources destinées aux organisations comptant des milliers de comptes.

Pour plus d'informations, consultez Surveillance de la sécurité des données grâce à des services AWS de sécurité gérés.

Bonnes pratiques de surveillance et d'audit pour Amazon S3

Les bonnes pratiques suivantes pour Amazon S3 peuvent vous aider à détecter les vulnérabilités et les incidents de sécurité potentiels.

Identification et audit de tous vos compartiments Amazon S3

L'identification de vos ressources informatiques est un aspect crucial de la gouvernance et de la sécurité. Vous devez avoir une visibilité sur toutes vos ressources Amazon S3 pour évaluer leur niveau de sécurité et agir sur les zones de vulnérabilité potentielles. Pour auditer vos ressources, nous vous recommandons de procéder comme suit :

- Utilisez Tag Editor pour identifier et baliser les ressources sensibles en matière de sécurité ou d'audit, puis utilisez ces identifications pour rechercher ces ressources. Pour plus d'informations, consultez la section Recherche de ressources à étiqueter dans le Guide de l'utilisateur AWS des ressources de balisage.
- Utilisez l'inventaire S3 pour auditer et signaler le statut de réplication et de chiffrement de vos objets à des fins professionnelles, de conformité et d'obligations réglementaires. Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).
- Créez des groupes de ressources pour vos ressources Amazon S3. Pour plus d'informations, consultez Que sont les groupes de ressources ? dans le Guide de l'utilisateur AWS Resource Groups .

Mettre en œuvre la surveillance à l'aide d'outils AWS de surveillance

La surveillance joue un rôle important dans le maintien de la fiabilité, de la sécurité, de la disponibilité et des performances d'Amazon S3 et de vos AWS solutions. AWS fournit plusieurs

outils et services pour vous aider à surveiller Amazon S3 et vos autres Services AWS. Par exemple, vous pouvez surveiller CloudWatch les métriques Amazon pour Amazon S3, en particulier les `DeleteRequests` métriques `PutRequests` `GetRequests4xxErrors`,, et. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch et Surveillance d'Amazon S3.

Pour obtenir un deuxième exemple, veuillez consulter Exemple : Activité de compartiment Amazon S3. Cet exemple décrit comment créer une CloudWatch alarme qui est déclenchée lorsqu'un appel d'API Amazon S3 est envoyé à une politique de compartiment, à `PUT` `DELETE` un cycle de vie de compartiment, à une configuration de réplication de compartiment ou à `PUT` une ACL de compartiment.

Activation de la journalisation des accès au serveur Amazon S3

La journalisation des accès au serveur fournit des enregistrements détaillés sur les demandes soumises à un compartiment. Les journaux d'accès au serveur peuvent vous aider pour les audits de sécurité et d'accès, et vous permettre d'en savoir plus sur votre base de clients et de comprendre votre facture Amazon S3. Pour obtenir des instructions pour l'activation de la journalisation des accès au serveur, veuillez consulter Enregistrement de demandes avec journalisation des accès au serveur.

Envisagez également de mettre en œuvre des contrôles de détection continus à l'aide de la règle s3-bucket-logging-enabled AWS Config gérée.

Utilisez AWS CloudTrail

AWS CloudTrail fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS dans Amazon S3. Vous pouvez utiliser les informations collectées par CloudTrail pour déterminer les éléments suivants :

- La demande qui a été adressée à Amazon S3
- L'adresse IP à partir de laquelle la demande a été effectuée
- La personne ayant effectué la demande
- La date et l'heure où la demande a été effectuée
- Des détails supplémentaires sur la demande

Par exemple, vous pouvez identifier les CloudTrail entrées pour les `PUT` actions qui affectent l'accès aux données`PutBucketAcl`, en particulier`PutObjectAcl`,`PutBucketPolicy`, et`PutBucketWebsite`.

Lorsque vous configurez votre Compte AWS, CloudTrail est activé par défaut. Vous pouvez consulter les événements récents dans la CloudTrail console. Pour créer un enregistrement continu de l'activité et des événements de vos compartiments Amazon S3, vous pouvez créer un suivi dans la CloudTrail console. Pour plus d'informations, consultez Journalisation des événements de données dans le Guide de l'utilisateur AWS CloudTrail .

Lorsque vous créez un suivi, vous pouvez le configurer CloudTrail pour consigner les événements liés aux données. Les événements de données sont des enregistrements d'opérations de ressources effectuées sur ou dans une ressource. Dans Amazon S3, les événements de données enregistrent l'activité de l'API au niveau de l'objet pour des compartiments individuels. CloudTrail prend en charge un sous-ensemble d'opérations d'API au niveau des objets Amazon S3, telles que`GetObject`, et`DeleteObject`. `PutObject` Pour plus d'informations sur le CloudTrail fonctionnement d'Amazon S3, consultezJournalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail. Dans la console Amazon S3, vous pouvez également configurer vos compartiments S3 pour Activation de la journalisation des CloudTrail événements pour les compartiments et les objets S3.

AWS Config fournit une règle gérée (`cloudtrail-s3-dataevents-enabled`) que vous pouvez utiliser pour vérifier qu'au moins un journal CloudTrail enregistre des événements de données pour vos compartiments S3. Pour plus d’informations, consultez cloudtrail-s3-dataevents-enabled dans le Guide du développeur AWS Config .

Activer AWS Config

Plusieurs des meilleures pratiques répertoriées dans cette rubrique suggèrent de créer des AWS Config règles. AWS Config vous aide à évaluer, auditer et évaluer les configurations de vos AWS ressources. AWS Config surveille les configurations des ressources afin que vous puissiez évaluer les configurations enregistrées par rapport aux configurations sécurisées souhaitées. Avec AWS Config, vous pouvez effectuer les opérations suivantes :

- Examiner les modifications apportées aux configurations et aux relations entre les ressources AWS
- Suivre les historiques détaillés de la configuration des ressources
- Déterminer votre conformité globale face aux configurations spécifiées dans vos instructions internes

Son utilisation AWS Config peut vous aider à simplifier l'audit de conformité, l'analyse de sécurité, la gestion des modifications et le dépannage opérationnel. Pour plus d'informations, consultez la

section Configuration AWS Config avec la console dans le guide du AWS Config développeur. Lors de la spécification des types de ressources à enregistrer, assurez-vous d'inclure les ressources Amazon S3.

> **Important**
>
> AWS Config les règles gérées ne prennent en charge que les compartiments à usage général lors de l'évaluation des ressources Amazon S3. AWS Config n'enregistre pas les modifications de configuration pour les compartiments de répertoire. Pour plus d'informations, consultez les AWS Config sections Règles gérées et Liste des règles AWS Config gérées dans le Guide du AWS Config développeur.

Pour un exemple d'utilisation AWS Config, consultez Comment utiliser AWS Config pour surveiller et répondre aux compartiments Amazon S3 autorisant un accès public sur le blog sur la AWS sécurité.

Découverte de données sensibles à l'aide d'Amazon Macie

Amazon Macie est un service de sécurité qui découvre les données sensibles au moyen du machine learning et de la correspondance de modèles. Macie fournit une visibilité sur les risques liés à la sécurité des données et permet une protection automatisée contre ces risques. Grâce à Macie, vous pouvez automatiser la découverte et la création de rapports de données sensibles dans votre parc de données Amazon S3 afin de mieux comprendre les données stockées par votre organisation dans S3.

Pour détecter les données sensibles avec Macie, vous pouvez utiliser des critères et des techniques intégrés, conçus pour détecter une liste étendue et croissante de types de données sensibles pour de nombreux pays et régions. Ces types de données sensibles incluent plusieurs types de données d'identification personnelle (PII), des données financières et des informations d'identification. Vous pouvez également utiliser des critères personnalisés que vous définissez : des expressions régulières qui définissent des modèles de texte à mettre en correspondance et, éventuellement, des séquences de caractères et des règles de proximité pour affiner les résultats.

Si Macie détecte des données sensibles dans un objet S3, Macie génère un résultat de sécurité pour vous en informer. Ce résultat fournit des informations sur l'objet affecté, les types et le nombre d'occurrences des données sensibles découvertes par Macie, ainsi que des détails supplémentaires pour vous aider à mener des investigations sur le compartiment et l'objet S3 affectés. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon Macie.

Utilisation de S3 Storage Lens

S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

Avec S3 Storage Lens, vous pouvez utiliser des métriques pour générer des informations récapitulatives, telles que la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes qui connaissent la croissance la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier des opportunités d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection des données et de gestion des accès, et améliorer les performances des charges de travail d'application.

Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour abandonner les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3. Pour plus d'informations, consultez la section Présentation d'Amazon S3 Storage Lens.

Surveillance des conseils de sécurité AWS

Nous vous recommandons de vérifier régulièrement les conseils de sécurité publiés dans Trusted Advisor pour votre Compte AWS. En particulier, recherchez des avertissements concernant les compartiments Amazon S3 avec des autorisations permettant un libre accès. Vous pouvez procéder par programmation en utilisant describe-trusted-advisor-checks.

De plus, surveillez activement l'adresse e-mail principale enregistrée pour chacun de vos Comptes AWS. AWS utilise cette adresse e-mail pour vous contacter au sujet de problèmes de sécurité émergents susceptibles de vous affecter.

AWS les problèmes opérationnels ayant un large impact sont publiés sur le site AWS Health Dashboard - Santé du service. Les problèmes opérationnels sont également publiés dans les comptes individuels via le tableau de bord AWS Health Dashboard. Pour en savoir plus, consultez la documentation AWS Health.

Surveillance de la sécurité des données grâce à des services AWS de sécurité gérés

Plusieurs services AWS de sécurité gérés peuvent vous aider à identifier, évaluer et surveiller les risques de sécurité et de conformité pour vos données Amazon S3. Ils peuvent également vous aider à protéger vos données contre ces risques. Ces services incluent des fonctionnalités de détection, de surveillance et de protection automatisées conçues pour passer des ressources Amazon S3 individuelles Compte AWS aux ressources destinées aux entreprises comptant des milliers de personnes Comptes AWS.

AWS les services de détection et de réponse peuvent vous aider à identifier les erreurs de configuration de sécurité, les menaces ou les comportements inattendus potentiels, afin que vous puissiez réagir rapidement à des activités potentiellement non autorisées ou malveillantes dans votre environnement. AWS les services de protection des données peuvent vous aider à surveiller et à protéger vos données, vos comptes et vos charges de travail contre tout accès non autorisé. Ils peuvent également vous aider à découvrir des données sensibles, telles que des données d'identification personnelle (PII), dans votre parc de données Amazon S3.

Pour vous aider à identifier et à évaluer les risques liés à la sécurité et à la conformité des données, les services de sécurité AWS gérés génèrent des résultats pour vous informer des événements ou des problèmes de sécurité potentiels liés à vos données Amazon S3. Ces résultats fournissent des informations pertinentes qui vous serviront à étudier et évaluer ces risques et à y réagir en fonction de vos flux de travail et de vos politiques de réponse aux incidents. Vous pouvez accéder directement aux données des résultats en utilisant chaque service. Vous pouvez également envoyer les données à d'autres applications, services et systèmes, tels que votre système de gestion des incidents et des événements de sécurité (SIEM).

Pour surveiller la sécurité de vos données Amazon S3, pensez à utiliser ces services AWS de sécurité gérés.

Amazon GuardDuty

Amazon GuardDuty est un service de détection des menaces qui surveille en permanence votre activité Comptes AWS et votre charge de travail pour détecter toute activité malveillante et fournit des résultats de sécurité détaillés à des fins de visibilité et de correction.

Avec la fonctionnalité de protection S3 intégrée GuardDuty, vous pouvez configurer GuardDuty pour analyser les événements AWS CloudTrail de gestion et de données relatifs à vos ressources

Amazon S3. GuardDuty surveille ensuite ces événements pour détecter toute activité malveillante et suspecte. Pour éclairer l'analyse et identifier les risques de sécurité potentiels, GuardDuty utilise des flux de renseignements sur les menaces et l'apprentissage automatique.

GuardDuty peut surveiller différents types d'activité pour vos ressources Amazon S3. Par exemple, les événements CloudTrail de gestion pour Amazon S3 incluent des opérations au niveau du compartiment, telles que `ListBucketsDeleteBucket`, et. `PutBucketReplication` CloudTrail les événements de données pour Amazon S3 incluent des opérations au niveau de l'objet, telles que `GetObjectListObjects`, et. `PutObject` S'il GuardDuty détecte une activité anormale ou potentiellement malveillante, il génère une constatation pour vous en informer.

Pour plus d'informations, consultez Amazon S3 Protection dans Amazon GuardDuty dans le guide de GuardDuty l'utilisateur Amazon.

Amazon Detective

Amazon Detective simplifie le processus d'investigation et vous aide à mener des investigations de sécurité plus rapides et plus efficaces. Detective fournit des agrégations de données, des résumés et un contexte prédéfinis qui peuvent vous aider à analyser et à évaluer la nature et l'étendue des éventuels problèmes de sécurité.

Detective extrait automatiquement les événements temporels, tels que les appels d'API AWS CloudTrail et les journaux de flux Amazon VPC pour AWS vos ressources. Il ingère également les résultats générés par Amazon GuardDuty. Detective utilise ensuite le machine learning, l'analyse statistique et la théorie des graphes pour générer des visualisations qui vous aideront à mener plus rapidement des investigations de sécurité efficaces.

Ces visualisations fournissent une vue unifiée et interactive des comportements des ressources et de leurs interactions au fil du temps. Vous pouvez explorer ce graphique de comportement pour examiner les actions potentiellement malveillantes, telles que les tentatives de connexion infructueuses ou les appels d'API suspects. Vous pouvez également voir comment ces actions affectent les ressources, notamment les objets et les compartiments S3.

Pour plus d'informations, consultez le Guide d'administration Amazon Detective.

IAM Access Analyzer

AWS Identity and Access Management Access Analyzer (IAM Access Analyzer) peut vous aider à identifier les ressources partagées avec une entité externe. Vous pouvez également utiliser

IAM Access Analyzer pour valider les politiques IAM par rapport à la grammaire des politiques et aux meilleures pratiques, et générer des politiques IAM en fonction de l'activité d'accès dans vos journaux. AWS CloudTrail

IAM Access Analyzer utilise un raisonnement basé sur la logique pour analyser les politiques de ressources de votre AWS environnement, telles que les politiques relatives aux compartiments. Avec IAM Access Analyzer pour S3, vous êtes alerté lorsqu'un compartiment S3 est configuré pour autoriser l'accès à toute personne sur Internet ou autre Comptes AWS, y compris à des comptes extérieurs à votre organisation. Par exemple, l'analyseur d'accès IAM pour S3 peut signaler qu'un compartiment dispose d'un accès en lecture ou en écriture fourni via une liste de contrôle d'accès (ACL) de compartiment, une politique de compartiment, une politique de point d'accès multirégion ou une politique de point d'accès. Pour chaque compartiment public ou partagé, vous recevez des résultats qui indiquent la source et le niveau d'accès public ou partagé. Grâce à ces résultats, vous pouvez prendre des mesures correctives immédiates et précises pour restaurer l'accès au compartiment comme vous l'aviez prévu.

Pour plus d’informations, consultez Examen de l'accès aux compartiments à l'aide de l'analyseur d'accès IAM pour S3.

Amazon Macie

Amazon Macie est un service de sécurité des données qui découvre les données sensibles à l'aide du machine learning et de la correspondance de modèles, fournit une visibilité sur les risques liés à la sécurité des données et permet une protection automatisée contre ces risques.

Grâce à Macie, vous pouvez automatiser la découverte et la création de rapports de données sensibles dans vos compartiments S3 afin de mieux comprendre les données stockées par votre organisation dans Amazon S3. Pour détecter les données sensibles, vous pouvez utiliser des critères et des techniques intégrés fournis par Macie, des critères personnalisés que vous définissez ou une combinaison des deux. Si Macie détecte des données sensibles dans un objet S3, Macie génère un résultat pour vous en informer. Ce résultat fournit des informations sur l'objet et le compartiment affectés, les types et le nombre d'occurrences des données sensibles découvertes par Macie, ainsi que des détails supplémentaires pour vous aider à mener des investigations.

Macie fournit également des statistiques et d'autres données qui offrent un insight de la posture de sécurité de vos données Amazon S3, et il évalue et surveille automatiquement vos compartiments S3 pour la sécurité et le contrôle d'accès. Si Macie détecte un problème potentiel lié à la sécurité ou la confidentialité de vos données, tel qu'un compartiment devenant accessible

au public, il génère un résultat que vous devrez examiner et auquel vous pourrez remédier si nécessaire.

Pour plus d'informations, consultez le Guide de l'utilisateur Amazon Macie.

AWS Security Hub

AWS Security Hub est un service de gestion de la posture de sécurité qui effectue des vérifications des meilleures pratiques de sécurité, regroupe les alertes et les résultats provenant de sources multiples dans un format unique et permet des mesures correctives automatisées.

Security Hub collecte et fournit des données relatives aux résultats de AWS Partner Network sécurité à partir de solutions de sécurité intégrées Services AWS, notamment Amazon Detective, Amazon GuardDuty, IAM Access Analyzer et Amazon Macie. Il génère également ses propres conclusions en effectuant des contrôles de sécurité continus et automatisés basés sur les AWS meilleures pratiques et les normes du secteur prises en charge.

Security Hub met ensuite en corrélation et consolide les résultats entre les fournisseurs, afin de vous aider à hiérarchiser et à traiter les résultats les plus significatifs. Il prend également en charge les actions personnalisées, que vous pouvez utiliser pour invoquer des réponses ou des actions de correction pour des classes spécifiques de résultats.

Avec Security Hub, vous pouvez évaluer l'état de sécurité et de conformité de vos ressources Amazon S3, dans le cadre d'une analyse plus large du niveau de sécurité de votre entreprise au niveau individuel Régions AWS et dans plusieurs régions. Cela inclut l'analyse des tendances en matière de sécurité et l'identification des problèmes de sécurité prioritaires. Vous pouvez également agréger les résultats de plusieurs Régions AWS, ainsi que surveiller et traiter les données de résultats agrégées provenant d'une seule région.

Pour plus d'informations, consultez Contrôles Amazon Simple Storage Service dans le Guide de l'utilisateur AWS Security Hub .

Gestion de votre stockage Amazon S3

Une fois que vous avez créé des compartiments et chargé des objets dans Simple Storage Service (Amazon S3), vous pouvez gérer votre stockage d'objets à l'aide de fonctionnalités telles que la gestion des versions, les classes de stockage, le verrouillage des objets, les opérations par lot, la réplication, les balises, etc. Les sections suivantes fournissent des informations détaillées sur les capacités et fonctionnalités de gestion du stockage disponibles dans Amazon S3.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Utilisation de la gestion des versions dans les compartiments S3

La gestion des versions Simple Storage Service (Amazon S3) permet de conserver plusieurs variantes d'un objet dans le même compartiment. Vous pouvez utiliser la fonctionnalité de gestion des versions S3 pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans vos compartiments. La gestion des versions permet de récupérer facilement les données en cas d'action involontaire d'un utilisateur ou de défaillance applicative. Lorsque la gestion des versions est activée pour un compartiment, si Simple Storage Service (Amazon S3) reçoit simultanément plusieurs demandes d'écriture pour le même objet, il stocke tous ces objets.

Les compartiments activés pour la gestion des versions vous permettent de récupérer des objets en cas de suppression ou remplacement accidentel. Par exemple, si vous supprimez un objet, Amazon S3 insère un marqueur de suppression au lieu de supprimer l'objet définitivement. Le marqueur de suppression devient la version actuelle de l'objet. Si vous remplacez un objet, cela crée une nouvelle version d'objet dans le compartiment. Vous pouvez toujours restaurer la version précédente. Pour plus d'informations, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.

Par défaut, la gestion des versions S3 est désactivé sur les compartiments et vous devez l'activer explicitement. Pour plus d'informations, consultez Activation de la gestion des versions sur les compartiments.

> **Note**
>
> - L'API SOAP ne prend pas en charge la gestion des versions S3. La prise en charge de SOAP via HTTP est obsolète, mais continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP.
> - Les tarifs Amazon S3 normaux s'appliquent pour chaque version d'un objet stocké ou transféré. Chaque version d'un objet est l'objet entier ; il ne s'agit pas simplement d'une différence par rapport à la version précédente. Par conséquent, si vous possédez trois versions d'un objet stocké, vous êtes facturé pour trois objets.

Compartiments non versionnés, activés pour la gestion des versions et suspendus

Les compartiments peuvent être dans l'un des trois états suivants :

- Non versionné (valeur par défaut)
- Gestion des versions activée
- Gestion des versions suspendue

L'activation et la suspension de la gestion des versions se fait au niveau du compartiment. Lorsque vous activez un compartiment pour la gestion des versions, il ne peut jamais revenir à un état non versionné. Toutefois, il est possible d'annuler la gestion des versions sur ce compartiment.

L'état de gestion des versions s'applique à tous les objets (jamais à certains) du compartiment. Lorsque vous activez la gestion des versions dans un compartiment, tous les nouveaux objets sont versionnés et reçoivent un ID de version unique. Les objets qui existaient déjà dans le compartiment au moment où la gestion de versions a été activée seront toujours versionnés et dotés d'un ID de version unique lorsqu'ils seront modifiés par des demandes futures. Remarques :

- Les objets qui sont stockés dans le compartiment avant que vous définissiez l'état de la gestion des versions ont un ID de version `null`. Lorsque vous activez la gestion des versions, les objets existants dans le compartiment ne changent pas. Seule la façon dont Amazon S3 gère les objets dans les futures demandes change. Pour plus d’informations, consultez Utiliser des objets dans un compartiment activé pour la gestion des versions.
- Le propriétaire du compartiment (ou tout utilisateur doté des autorisations adaptées) peut désactiver la gestion des versions pour stopper l'accumulation des versions d'objet. Lorsque vous désactivez la gestion des versions, les objets existants du compartiment ne changent pas. Seule la façon dont Amazon S3 gère les objets dans les futures demandes change. Pour plus d’informations, consultez Utilisation des objets dans un compartiment désactivé pour la gestion des versions.

Utilisation de la gestion des versions S3 avec le cycle de vie S3

Pour personnaliser votre approche de la conservation des données et maîtriser les coûts de stockage, utilisez la gestion des versions des objets avec le cycle de vie S3. Pour plus d’informations, consultez Gestion du cycle de vie de votre stockage. Pour plus d'informations sur la création de configurations S3 Lifecycle à l' AWS Management Console aide AWS des SDK ou de l'API REST, consultezConfiguration du cycle de vie d'un bucket. AWS CLI

> **Important**
>
> Si vous avez une configuration de cycle de vie d'expiration des objets dans votre compartiment non versionné et que vous souhaitez conserver le même comportement de suppression définitive lorsque vous activez la gestion des versions, vous devez ajouter une configuration d'expiration des versions anciennes. La configuration de cycle de vie d'expiration des versions anciennes gère les suppressions des versions anciennes des objets dans le compartiment activé pour la gestion des versions. (Un compartiment activé pour la gestion des versions conserve une version d'objet actuelle et aucune ou plusieurs versions d'objet anciennes.) Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket.

Pour en savoir plus sur l'utilisation de la gestion des versions S3, veuillez consulter les rubriques suivantes.

Rubriques

- Fonctionnement de la gestion des versions S3
- Activation de la gestion des versions sur les compartiments
- Configuration de la fonction Supprimer MFA
- Utiliser des objets dans un compartiment activé pour la gestion des versions
- Utilisation des objets dans un compartiment désactivé pour la gestion des versions

Fonctionnement de la gestion des versions S3

Vous pouvez utiliser la gestion des versions S3 pour conserver plusieurs versions d'un objet dans un même compartiment afin de pouvoir restaurer des objets qui sont accidentellement supprimés ou remplacés. Par exemple, si vous appliquez la gestion des versions S3 à un compartiment, les modifications suivantes se produisent :

- Si vous supprimez un objet, au lieu de supprimer l'objet définitivement, Amazon S3 insère un marqueur de suppression, qui devient la version d'objet actuelle. Vous pouvez ensuite restaurer la version précédente. Pour plus d'informations, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.

- Si vous remplacez un objet, Amazon S3 ajoute une nouvelle version d'objet dans le compartiment. La version précédente reste dans le compartiment et devient une version ancienne. Vous pouvez restaurer la version précédente.

> **Note**
>
> Les tarifs Amazon S3 normaux s'appliquent pour chaque version d'un objet stocké et transféré. Chaque version d'un objet est l'objet entier ; il n'est pas différent de la version précédente. Par conséquent, si vous possédez trois versions d'un objet stocké, vous êtes facturé pour trois objets.

Chaque compartiment S3 créé possède une sous-ressource de gestion des version qui lui est associée. (Pour plus d'informations, consultez Options de configuration des compartiments.) Par défaut, votre compartiment est non versionné, et la sous-ressource de la gestion des versions stocke une configuration de gestion des versions vide, comme suit :

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Pour activer la gestion des versions, vous pouvez envoyer une demande à Amazon S3 avec une configuration de la gestion des versions qui inclut un statut `Enabled`.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Pour désactiver la gestion des versions, vous configurez la valeur de l'état sur `Suspended`.

> **Note**
>
> Quand vous activez la gestion des versions sur un compartiment pour la première fois, la propagation complète de la modification peut prendre un court laps de temps. Nous vous recommandons d'attendre 15 minutes après l'activation de la gestion des versions avant d'exécuter des opérations d'écriture (`PUT` ou `DELETE`) sur les objets du compartiment.

Le propriétaire du bucket et tous les utilisateurs autorisés AWS Identity and Access Management (IAM) peuvent activer le versionnement. Le propriétaire du bucket est celui Compte AWS qui a créé le bucket. Pour plus d'informations sur les autorisations, consultez Identity and Access Management pour Amazon S3.

Pour plus d'informations sur l'activation et la désactivation de la gestion des versions S3 à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou REST, consultez. the section called "Activation de la gestion des versions sur les compartiments"

Rubriques

- ID de version
- Flux de travail de la gestion des versions

ID de version

Si vous activez la gestion des versions pour un compartiment, Simple Storage Service (Amazon S3) génère automatiquement un ID de version unique pour l'objet qui est stocké. Par exemple, dans un compartiment, vous pouvez avoir deux objets avec la même clé (nom d'objet), mais des ID de version différents, comme `photo.gif` (version 111111) et `photo.gif` (version 121212).
Schéma illustrant un compartiment activé pour la gestion des versions comportant deux objets dotés de la même clé mais d'identifiants de version différents.

Chaque objet possède un ID de version, que la gestion des versions S3 soit activée ou non. Si la gestion des versions S3 n'est pas activée, Amazon S3 définit la valeur de l'ID de version sur `null`. Si vous activez la gestion des versions S3, Simple Storage Service (Amazon S3) attribue une valeur d'ID de version à l'objet. Cette valeur distingue l'objet des autres versions de la même clé.

Lorsque vous activez la gestion des versions S3 dans un compartiment existant, les objets déjà stockés dans le compartiment ne changent pas. Leurs ID de version (`null`), le contenu et les autorisations restent les mêmes. Après avoir activé la gestion des versions S3, chaque objet ajouté au compartiment obtient un ID de version, qui le distingue des autres versions de la même clé.

Seul Amazon S3 génère des ID de version, et ils ne peuvent pas être modifiés. Les ID de version sont des chaînes de caractères opaques Unicode, encodées UTF-8, prêtes pour l'URL, d'une longueur maximale de 1 024 octets. Voici un exemple :

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

> **Note**
>
> Pour plus de simplicité, les autres exemples de cette rubrique utilisent des ID beaucoup plus courts.

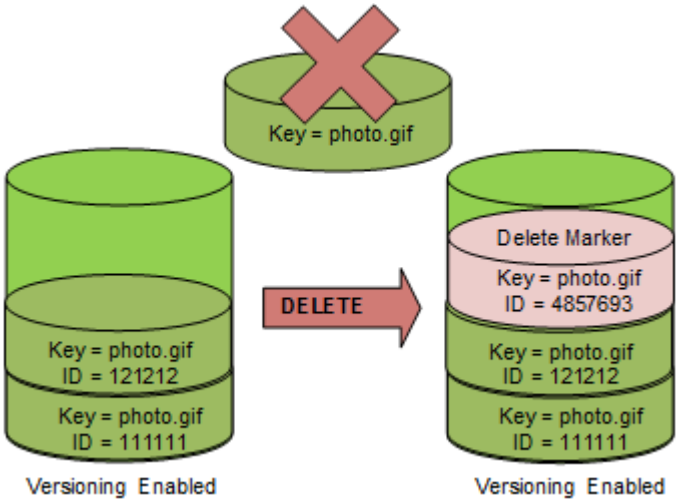
Flux de travail de la gestion des versions

Lorsque vous faites une demande PUT sur un objet dans un compartiment activé pour la gestion des versions, l'ancienne version n'est pas remplacée. Comme le montre la figure suivante, lorsqu'une nouvelle version de `photo.gif` est PUT dans un compartiment qui contient déjà un objet portant le même nom, le comportement suivant se produit :

- L'objet d'origine (ID = 111111) reste dans le compartiment.
- Amazon S3 génère un nouvel ID de version (121212) et ajoute cette nouvelle version de l'objet au compartiment.

Cette fonctionnalité vous permet de récupérer une version précédente d'un objet si celui-ci a été accidentellement remplacé ou supprimé.

Lorsque vous faites une requête DELETE sur un objet, toutes les versions restent dans le compartiment et Amazon S3 insère un marqueur de suppression, comme illustré dans le schéma suivant.



Le marqueur de suppression devient la version actuelle de l'objet. Par défaut, les demandes GET récupèrent la version la plus récemment stockée. Le fait d'effectuer une requête `GET Object` lorsque la version actuelle est un marqueur de suppression renvoie une erreur `404 Not Found`, comme illustré dans la figure suivante.

Vous pouvez, toutefois, vous pouvez faire une demande GET sur une ancienne version d'un objet en spécifiant son ID de version. Dans le schéma suivant, vous faites une demande GET sur une version d'objet spécifique, 111111. Amazon S3 renvoie une version d'objet même s'il ne s'agit pas de la version actuelle.

Pour plus d'informations, consultez Récupération de versions d'objets à partir d'un compartiment activé pour la gestion des versions.

Vous pouvez supprimer définitivement un objet en spécifiant la version que vous souhaitez supprimer. Seul le propriétaire d'un compartiment Amazon S3 ou un utilisateur IAM autorisé peut supprimer définitivement une version. Si votre opération DELETE inclut l'élément `versionId`, cette version de l'objet est définitivement supprimée et Amazon S3 n'insère pas de marqueur de suppression.

Vous pouvez ajouter une sécurité supplémentaire en configurant un compartiment pour activer la suppression par authentification multifactorielle (MFA). Quand vous activez la suppression MFA pour un compartiment, le propriétaire du compartiment doit inclure deux formes d'authentification pour toute demande de suppression d'une version ou de changement de l'état de la gestion des versions du compartiment. Pour plus d'informations, consultez Configuration de la fonction Supprimer MFA.

Quand les nouvelles versions sont-elles créées pour un objet ?

Les nouvelles versions sont créées uniquement lorsque vous faites une requête PUT sur un nouvel objet. Sachez que certaines actions, comme `CopyObject`, fonctionnent en mettant en œuvre une opération PUT.

Certaines actions qui modifient l'objet actuel ne créent pas de nouvelle version, car elles ne font pas de requête PUT sur le nouvel objet. Cela inclut des actions telles que la modification des balises sur un objet.

> **Important**
>
> Si vous remarquez une augmentation importante du nombre de réponses HTTP 503 (Service indisponible) reçues pour les requêtes d'objet PUT ou DELETE Amazon S3 concernant un

compartiment pour lequel la gestion des versions S3 est activée, il est possible qu'il y ait un ou plusieurs objets du compartiment avec des millions de versions. Pour plus d'informations, consultez la section sur la gestion des versions S3 de Résolution des problèmes.

Activation de la gestion des versions sur les compartiments

Vous pouvez utiliser la gestion des versions Simple Storage Service (Amazon S3) pour conserver plusieurs versions d'un objet dans un même compartiment. Cette section fournit des exemples expliquant comment activer le versionnement sur un bucket à l'aide de la console, de l'API REST, AWS des SDK et AWS Command Line Interface ()AWS CLI.

> **Note**
>
> Si vous activez le versionnement sur un bucket pour la première fois, la propagation complète de la modification peut prendre jusqu'à 15 minutes. Nous vous recommandons d'attendre 15 minutes après l'activation de la gestion des versions avant d'exécuter des opérations d'écriture (PUT ou DELETE) sur les objets du compartiment. Les opérations d'écriture effectuées avant la fin de cette conversion peuvent s'appliquer à des objets non versionnés.

Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3. Pour en savoir plus sur l'utilisation d'objets se trouvant dans un compartiment dont la gestion des versions est activée, veuillez consulter Utiliser des objets dans un compartiment activé pour la gestion des versions.

Pour en savoir plus sur l'utilisation de la gestion des versions S3 pour protéger les données, consultez Tutoriel : protection des données sur Amazon S3 contre les suppressions accidentelles ou les bogues d'application à l'aide de l'archivage par versions S3, du verrouillage d'objets S3 et de la réplication S3.

Chaque compartiment S3 créé possède une sous-ressource de gestion des version qui lui est associée. (Pour plus d'informations, consultez Options de configuration des compartiments.) Par défaut, votre compartiment est non versionné, et la sous-ressource de la gestion des versions stocke une configuration de gestion des versions vide, comme suit :

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Pour activer la gestion des versions, vous pouvez envoyer une demande à Simple Storage Service (Amazon S3) avec une configuration de la gestion des versions qui inclut un statut.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Pour désactiver la gestion des versions, vous configurez la valeur de l'état sur `Suspended`.

Le propriétaire du compartiment et tous les utilisateurs autorisés peuvent activer la gestion des versions. Le propriétaire du bucket est celui Compte AWS qui a créé le bucket (le compte root). Pour plus d'informations sur les autorisations, consultez Identity and Access Management pour Amazon S3.

Les sections suivantes fournissent plus de détails sur l'activation de la gestion des versions S3 à l'aide de la console et des AWS SDK. AWS CLI

Utilisation de la console S3

Procédez comme suit pour utiliser le AWS Management Console afin d'activer le versionnement sur un compartiment S3.

Pour activer ou désactiver la gestion des versions dans un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer la gestion des versions.
3. Choisissez Propriétés.
4. Sous Bucket Versioning (Gestion des versions de compartiment), choisissez Edit (Modifier).
5. Choisissez Suspend (Interrompre) ou Enable (Activer), puis Save changes (Enregistrer les modifications).

> **Note**
>
> Vous pouvez utiliser l'authentification AWS multifactorielle (MFA) avec le versionnement. Lorsque vous utilisez l'authentification MFA avec le contrôle de version, vous devez fournir vos clés Compte AWS d'accès et un code valide provenant du dispositif MFA du compte pour supprimer définitivement une version d'objet ou suspendre ou réactiver le contrôle de version.

Pour utiliser l'authentification multi-facteurs (MFA) avec la gestion des versions, vous activez `MFA Delete`. Vous ne pouvez pas activer `MFA Delete` à l'aide de la AWS Management Console. Vous devez utiliser le AWS Command Line Interface (AWS CLI) ou l'API. Pour plus d'informations, consultez Configuration de la fonction Supprimer MFA.

En utilisant le AWS CLI

L'exemple suivant active la gestion des versions sur un compartiment S3.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled
```

L'exemple suivant active la gestion des versions S3 et la fonction Supprimer l'authentification multifactorielle (MFA) sur un compartiment.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Note

L'utilisation de la fonction Supprimer MFA nécessite un périphérique d'authentification physique ou virtuel approuvé. Pour en savoir plus sur l'utilisation de la fonction Supprimer MFA dans Amazon S3, veuillez consulter Configuration de la fonction Supprimer MFA.

Pour plus d'informations sur l'activation du versionnement à l'aide du AWS CLI, consultez put-bucket-versioningla référence des AWS CLI commandes.

Utilisation des AWS SDK

Les exemples suivants activent le contrôle de version sur un compartiment, puis récupèrent le statut de version à l'aide du AWS SDK for Java et du. AWS SDK for .NET Pour plus d'informations sur l'utilisation d'autres kits SDK AWS , consultez le Centre pour développeurs AWS.

.NET

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using System;
using Amazon.S3;
using Amazon.S3.Model;

namespace s3.amazon.com.docsamples
{
    class BucketVersioningConfiguration
    {
        static string bucketName = "*** bucket name ***";

        public static void Main(string[] args)
        {
            using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))
            {
                try
                {
                    EnableVersioningOnBucket(client);
                    string bucketVersioningStatus =
 RetrieveBucketVersioningConfiguration(client);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    if (amazonS3Exception.ErrorCode != null &&
                        (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId")
                        ||
                        amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))
                    {
                        Console.WriteLine("Check the provided AWS Credentials.");
                        Console.WriteLine(
                        "To sign up for service, go to http://aws.amazon.com/s3");
                    }
                    else
                    {
                        Console.WriteLine(
                         "Error occurred. Message:'{0}' when listing objects",
                         amazonS3Exception.Message);
                    }
```

```
                }
            }

            Console.WriteLine("Press any key to continue...");
            Console.ReadKey();
        }

        static void EnableVersioningOnBucket(IAmazonS3 client)
        {

                PutBucketVersioningRequest request = new PutBucketVersioningRequest
                {
                    BucketName = bucketName,
                    VersioningConfig = new S3BucketVersioningConfig
                    {
                        Status = VersionStatus.Enabled
                    }
                };

                PutBucketVersioningResponse response =
 client.PutBucketVersioning(request);
        }


        static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
        {
                GetBucketVersioningRequest request = new GetBucketVersioningRequest
                {
                    BucketName = bucketName
                };

                GetBucketVersioningResponse response =
 client.GetBucketVersioning(request);
                return response.VersioningConfig.Status;
            }
    }
}
```

Java

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;

public class BucketVersioningConfigurationExample {
    public static String bucketName = "*** bucket name ***";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {

            // 1. Enable versioning on the bucket.
         BucketVersioningConfiguration configuration =
           new BucketVersioningConfiguration().withStatus("Enabled");

  SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest
 =
     new SetBucketVersioningConfigurationRequest(bucketName,configuration);


 s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);

  // 2. Get bucket versioning configuration information.
  BucketVersioningConfiguration conf =
 s3Client.getBucketVersioningConfiguration(bucketName);
   System.out.println("bucket versioning configuration status:    " +
 conf.getStatus());

       } catch (AmazonS3Exception amazonS3Exception) {
           System.out.format("An Amazon S3 error occurred. Exception: %s",
 amazonS3Exception.toString());
       } catch (Exception ex) {
           System.out.format("Exception: %s", ex.toString());
       }
   }
```

```
}
```

Python

L'exemple de code Python suivant crée un compartiment Amazon S3, l'active pour la gestion des versions, et configure un cycle de vie qui fait expirer les versions d'objet anciennes après 7 jours.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
 lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
```

```
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )

    return bucket
```

Configuration de la fonction Supprimer MFA

Lorsque vous travaillez avec la gestion des versions S3 dans des compartiments Simple Storage Service (Amazon S3), vous pouvez ajouter une couche de sécurité en activant la fonction MFA delete (Suppression de l'authentification multifacteur). Quand vous procédez ainsi, le propriétaire du

compartiment doit inclure deux formes d'authentification dans toute demande pour supprimer une version ou modifier l'état de la gestion des versions du compartiment.

La fonction Supprimer MFA exige une authentification supplémentaire pour les opérations suivantes :

- Changer l'état de la gestion des versions de votre compartiment
- Supprimer définitivement une version d'objet

La fonction Supprimer MFA exige deux formes d'authentification appliquées ensemble :

- Les informations d'identification de sécurité
- L'enchaînement d'un numéro de série valide, d'un espace et du code à six chiffres affiché sur un appareil d'authentification approuvé

La fonction Supprimer MFA fournit donc une sécurité supplémentaire, par exemple, en cas de mise en danger de vos informations d'identification de sécurité. La fonction Supprimer MFA peut aider à prévenir les suppressions accidentelles de compartiments en obligeant l'utilisateur qui lance l'action de suppression à prouver la possession physique d'un appareil MFA avec un code MFA et en ajoutant une couche de friction et de sécurité supplémentaire à l'action de suppression.

Pour identifier les compartiments pour lesquels la suppression MFA est activée, vous pouvez utiliser les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec S3 Storage Lens. Pour obtenir la liste complète des métriques, consultez le Glossaire des métriques S3 Storage Lens.

Le propriétaire du bucket, celui Compte AWS qui a créé le bucket (compte root) et tous les utilisateurs autorisés peuvent activer le versionnement. Toutefois, seul le propriétaire du compartiment (compte racine) peut activer la fonction Supprimer MFA. Pour plus d'informations, consultez la section Sécurisation de l'accès à AWS l'utilisation de la MFA sur le blog de AWS sécurité.

> **Note**
>
> Pour utiliser la suppression MFA avec la gestion des versions, vous devez activer `MFA Delete`. Toutefois, vous ne pouvez pas activer `MFA Delete` à l'aide de la AWS

Management Console. Vous devez utiliser le AWS Command Line Interface (AWS CLI) ou l'API.

Consultez la section Exemples dans la rubrique Activation de la gestion des versions sur les compartiments pour découvrir des cas d'utilisation de la suppression MFA avec la gestion des versions.

Vous ne pouvez pas utiliser la suppression MFA avec des configurations de cycle de vie. Pour en savoir plus sur les configurations de cycle de vie et sur leur interaction avec d'autres configurations, consultez Cycle de vie et autres configurations de compartiment.

Pour activer ou désactiver la suppression MFA, vous devez utiliser la même API que celle utilisée pour configurer la gestion des versions sur un compartiment. Simple Storage Service (Amazon S3) stocke la configuration de la fonction Supprimer MFA dans la même sous-ressource de gestion des versions que celle de l'état de la gestion des versions du compartiment.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Pour utiliser la fonction Supprimer MFA, vous pouvez utiliser un appareil MFA matériel ou virtuel pour générer un code d'authentification. L'exemple suivant montre un code d'authentification généré et affiché sur un périphérique matériel.

Les fonctions Supprimer MFA et d'accès à l'API protégée par MFA sont conçues pour fournir une protection dans différents scénarios. Vous configurez la fonction Supprimer MFA sur un compartiment pour garantir que les données de votre compartiment ne puissent pas être supprimées par erreur. La fonction d'accès à l'API protégée par MFA est utilisée pour appliquer un autre facteur d'authentification (code MFA) lors de l'accès aux ressources Amazon S3 sensibles. Vous pouvez exiger que toute opération sur ces ressources Amazon S3 soit effectuée avec des informations d'identification temporaires créées grâce à la fonction MFA. Pour voir un exemple, consultez Exigence d'une MFA.

Pour en savoir plus sur l'achat et l'activation d'un appareil d'authentification, veuillez consulter la section Authentification multi-facteurs.

Pour activer la gestion des versions S3 et configurer la fonction Supprimer MFA

En utilisant le AWS CLI

L'exemple suivant active la gestion des versions S3 et la fonction Supprimer l'authentification multifactorielle (MFA) sur un compartiment.

```
aws s3api put-bucket-versioning --bucket example-s3-bucket1 --versioning-configuration
 Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Utilisation de l'API REST

Pour plus d'informations sur la spécification de la suppression MFA à l'aide de l'API REST Amazon S3, consultez le manuel PutBucketVersioningAmazon Simple Storage Service API Reference.

Utiliser des objets dans un compartiment activé pour la gestion des versions

Les objets qui sont stockés dans un compartiment Simple Storage Service (Amazon S3) avant d'en définir l'état de la gestion des versions ont un ID de version `null`. Lorsque vous activez la gestion des versions, les objets existants dans le compartiment ne changent pas. Seule la façon dont Amazon S3 gère les objets dans les futures demandes change.

Transition des versions d'un objet

Vous pouvez définir les règles de configuration du cycle de vie pour les objets qui possèdent un cycle de vie bien défini, afin de transférer les versions d'objet vers la classe de stockage S3 Glacier Flexible Retrieval à un moment spécifique dans la durée de vie de l'objet. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.

Les rubriques de cette section expliquent plusieurs opérations d'objet dans un compartiment activé pour la gestion des versions. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3.

Rubriques

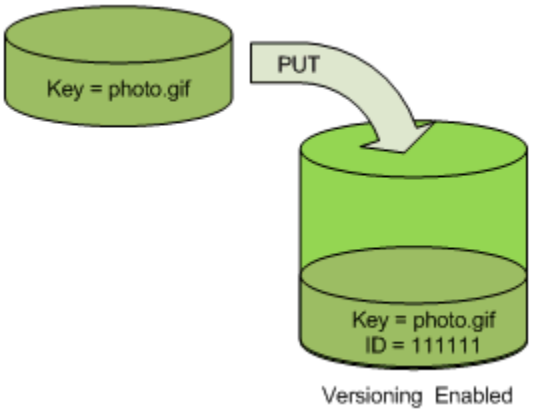
- Ajout d'objets dans des compartiments activés pour la gestion des versions
- Liste d'objets dans un compartiment activé pour la gestion des versions
- Récupération de versions d'objets à partir d'un compartiment activé pour la gestion des versions
- Suppression des versions d'objet d'un compartiment activé pour la gestion des versions

- Configuration des autorisations d'objet soumis à la gestion des versions

Ajout d'objets dans des compartiments activés pour la gestion des versions

Après avoir activé la gestion des versions sur un compartiment, Simple Storage Service (Amazon S3) ajoute automatiquement un ID de version unique à chaque objet stocké (à l'aide de `PUT`, `POST` ou `CopyObject`) dans le compartiment.

Le schéma suivant montre qu'Amazon S3 ajoute un ID de version unique à un objet ajouté à un compartiment activé pour la gestion des versions.



> **Note**
>
> Les valeurs de l'ID de version qu'Amazon S3 attribue sont sûres pour l'URL (elles peuvent être incluses dans une URI).

Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3. Vous pouvez ajouter des versions d'objets à un compartiment activé pour la gestion des versions à l'aide de la console, des AWS SDK et de l'API REST.

Utilisation de la console

Pour obtenir des instructions, consultez Chargement d'objets.

Utilisation de SDK AWS

Pour des exemples de téléchargement d'objets à l'aide AWS des SDK pour Java, .NET et PHP, consultez. Chargement d'objets Les exemples de chargement d'objets dans des compartiments

non versionnés et activés pour la gestion des versions sont identiques, même si dans le cas des compartiments activés pour la gestion des versions, Amazon S3 attribue un numéro de version. Sinon, le numéro de version est null.

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez le AWS Developer Center.

Utilisation de l'API REST

Pour ajouter des objets dans des compartiments activés pour la gestion des versions

1. Activez la gestion des versions sur un compartiment grâce à une demande `PutBucketVersioning`.

   Pour plus d'informations, veuillez consulter PutBucketVersioning dans la Référence d'API Amazon Simple Storage Service.
2. Envoyez une demande `PUT`, `POST`, ou `CopyObject` pour stocker un objet dans le compartiment.

Lorsque vous ajoutez un objet dans un compartiment activé pour la gestion des versions, Simple Storage Service (Amazon S3) renvoie l'ID de version de l'objet dans l'en-tête de la réponse `x-amz-version-id`, comme illustré dans l'exemple suivant :

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Liste d'objets dans un compartiment activé pour la gestion des versions

Cette section fournit des exemples de listes de versions d'objet à partir d'un compartiment activé pour la gestion des versions. Amazon S3 stocke les informations relatives aux versions d'objet dans la sous-ressource versions associée au compartiment. Pour plus d'informations, consultez Options de configuration des compartiments. Pour répertorier les objets d'un compartiment dont la gestion des versions est activée, vous avez besoin de l'autorisation `ListBucketVersions`.

Utilisation de la console S3

Procédez comme suit pour utiliser la console Amazon S3 pour afficher les différentes versions d'un objet.

Pour afficher différentes versions d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Pour afficher la liste des versions des objets du compartiment, choisissez le commutateur Show versions (Afficher les versions).

   Pour chaque version d'objet, la console affiche un ID de version unique, la date et l'heure auxquelles la version d'objet a été créée, et d'autres propriétés. (Les objets stockés dans le compartiment avant la configuration de l'état de la gestion des versions possèdent un ID de version null.)

   Pour répertorier les objets sans les versions, choisissez le commutateur List versions (Répertorier les versions).

Vous pouvez également afficher, télécharger et supprimer les versions d'un objet dans le panneau de présentation de l'objet sur la console. Pour plus d'informations, consultez Affichage d'une présentation d'un objet dans la console Amazon S3.

> **Note**
>
> Pour accéder aux versions d'objets antérieures à 300 versions, vous devez utiliser la AWS CLI ou l'URL de l'objet.

> **Important**
>
> Vous pouvez annuler la suppression d'un objet uniquement si celui-ci a été supprimé en tant que version la plus récente (version actuelle). Vous ne pouvez pas restaurer une version précédente d'un objet supprimé. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Utilisation des AWS SDK

Les exemples de cette section montrent comment récupérer une liste d'objets à partir d'un compartiment activé pour la gestion des versions. Chaque demande renvoie jusqu'à 1 000 versions, sauf si vous spécifiez un nombre inférieur. Si le compartiment contient plus de versions que cette limite, vous devez envoyer une série de demandes pour récupérer la liste de toutes les versions. Ce processus de renvoi des résultats dans des « pages » s'appelle pagination.

Pour illustrer le fonctionnement de la pagination, les exemples limitent chaque réponse à deux versions d'objet. Après avoir récupéré la première page de résultats, chaque exemple vérifie si la liste de versions a été tronquée. Si tel est le cas, l'exemple continue de récupérer les pages jusqu'à ce que toutes les versions aient été récupérées.

> **Note**
>
> Les exemples suivants fonctionnent également avec un compartiment qui n'est pas activé pour la gestion des versions ou pour des objets qui n'ont pas de versions individuelles. Dans ces cas de figure, Amazon S3 renvoie la liste d'objets avec l'ID de version `null`.

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez le AWS Developer Center.

Java

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();
```

```
            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                    .withBucketName(bucketName)
                    .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
 versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                            objectSummary.getKey(),
                            objectSummary.getVersionId());
                    numVersions++;
                }
                // Check whether there are more pages of versions to retrieve. If
                // there are, retrieve them. Otherwise, exit the loop.
                if (versionListing.isTruncated()) {
                    versionListing =
 s3Client.listNextBatchOfVersions(versionListing);
                } else {
                    break;
                }
            }
            System.out.println(numVersions + " object versions retrieved in " +
 numPages + " pages");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ListObjectsVersioningEnabledBucketTest
    {
        static string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main(string[] args)
        {
            s3Client = new AmazonS3Client(bucketRegion);
            GetObjectListWithAllVersionsAsync().Wait();
        }

        static async Task GetObjectListWithAllVersionsAsync()
        {
            try
            {
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    // You can optionally specify key name prefix in the request
                    // if you want list of object versions of a specific object.

                    // For this example we limit response to return list of 2
 versions.
                    MaxKeys = 2
                };
                do
```

```
                {
                    ListVersionsResponse response = await
 s3Client.ListVersionsAsync(request);
                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If response is truncated, set the marker to get the next
                    // set of keys.
                    if (response.IsTruncated)
                    {
                        request.KeyMarker = response.NextKeyMarker;
                        request.VersionIdMarker = response.NextVersionIdMarker;
                    }
                    else
                    {
                        request = null;
                    }
                } while (request != null);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Utilisation de l'API REST

Example — Liste de toutes les versions d'objet dans un compartiment

Pour répertorier toutes les versions de tous les objets d'un compartiment, vous utilisez la sous-ressource `versions` dans une demande `GET Bucket`. Amazon S3 peut récupérer 1 000 objets

maximum, et chaque version d'objet compte pleinement comme un objet. Par conséquent, si un compartiment contient deux clés (par exemple, `photo.gif` et `picture.jpg`), et que la première clé comporte 990 versions et la seconde 400 versions, une seule demande peut récupérer les 990 versions de `photo.gif` et uniquement les 10 versions les plus récentes de `picture.jpg`.

Amazon S3 renvoie des versions d'objet dans l'ordre dans lequel elles sont stockées ; les plus récemment stockées sont renvoyées en premier.

Dans une demande `GET Bucket`, incluez la sous-ressource `versions`.

```
GET /?versions HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Example — Récupération de toutes les versions d'une clé

Pour récupérer un sous-ensemble de versions d'objet, vous utilisez les paramètres de requête pour `GET Bucket`. Pour plus d'informations, consultez GET Bucket.

1. Configurez le paramètre `prefix` sur la clé de l'objet que vous souhaitez récupérer.
2. Envoyez une demande `GET Bucket` grâce à la sous-ressource `versions` et `prefix`.

   ```
   GET /?versions&prefix=objectName HTTP/1.1
   ```

Example — Récupération d'objets à l'aide d'un préfixe

L'exemple suivant récupère des objets dont la clé est ou commence par `myObject`.

```
GET /?versions&prefix=myObject HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Vous pouvez utiliser les autres paramètres de la demande pour récupérer un sous-ensemble de toutes les versions de l'objet. Pour plus d'informations, veuillez consulter GET Bucket dans la Référence d'API Amazon Simple Storage Service.

Example — Récupération d'une liste d'objets supplémentaires si la réponse est tronquée

Si le nombre d'objets qui peuvent être renvoyés dans une demande GET dépasse la valeur de `max-keys`, la réponse contient `<isTruncated>true</isTruncated>`, et inclut la première clé (dans `NextKeyMarker`) et le premier ID de version (dans `NextVersionIdMarker`) qui satisfont la demande, mais qui n'ont pas été renvoyés. Vous utilisez ces valeurs renvoyées comme la position de départ dans une demande suivante pour récupérer les objets supplémentaires qui satisfont la demande GET.

Utilisez le processus suivant pour récupérer des objets supplémentaires qui satisfont la demande `GET Bucket versions` originale à partir d'un compartiment. Pour plus d'informations sur `key-marker`, `version-id-marker`, `NextKeyMarker` et `NextVersionIdMarker`, consultez GET Bucket dans la référence de l'API Amazon Simple Storage Service.

Ces quelques réponses supplémentaires satisfont à la requête GET d'origine :

- Configurez la valeur de `key-marker` sur la clé renvoyée dans `NextKeyMarker` dans la réponse précédente.
- Configurez la valeur de `version-id-marker` sur l'ID de version renvoyé dans `NextVersionIdMarker` dans la réponse précédente.
- Envoyez une demande `GET Bucket versions` avec `key-marker` et `version-id-marker`.

Example — Récupération des objets commençant par une clé et un ID de version spécifiés

```
GET /?versions&key-marker=myObject&version-id-marker=298459348571 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

En utilisant le AWS CLI

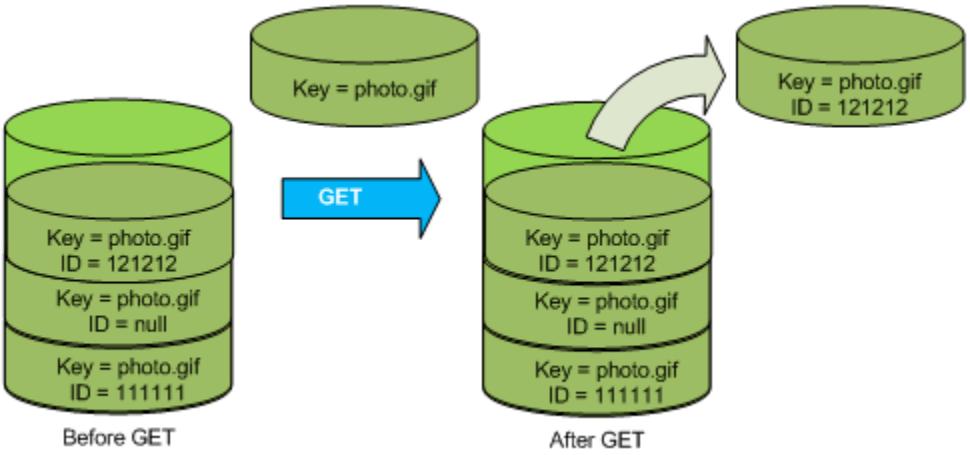
La commande suivante renvoie des métadonnées relatives à toutes les versions des objets contenus dans un compartiment.

```
aws s3api list-object-versions --bucket example-s3-bucket1
```

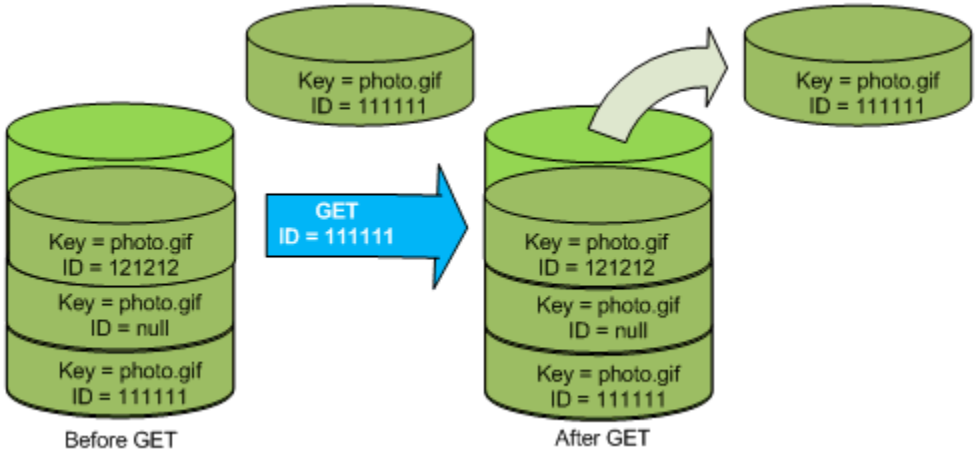
Pour plus d'informations sur `list-object-versions`, consultez list-object-versions dans la référence des commandes AWS CLI .

Récupération de versions d'objets à partir d'un compartiment activé pour la gestion des versions

La gestion des versions dans Simple Storage Service (Amazon S3) est un moyen de conserver plusieurs variantes d'un objet dans le même compartiment. Une simple demande `GET` récupère la version actuelle d'un objet. Le schéma suivant montre comment une demande `GET` renvoie la version actuelle de l'objet, `photo.gif`.



Pour récupérer une version spécifique, vous devez spécifier son ID de version. Le schéma suivant montre qu'une demande `GET versionId` récupère la version spécifiée de l'objet (pas nécessairement la version actuelle).



Vous pouvez récupérer des versions d'objets dans Amazon S3 à l'aide de la console, AWS des SDK ou de l'API REST.

> **Note**
>
> Pour accéder aux versions d'objets antérieures à 300 versions, vous devez utiliser la AWS CLI ou l'URL de l'objet.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, choisissez le nom de l'objet.
4. Choisissez Versions.

   Amazon S3 affiche toutes les versions de l'objet.
5. Sélectionnez la case à cocher en regard de l'ID de version des versions que vous souhaitez récupérer.
6. Choisissez Actions, choisissez Téléchargeret enregistrez l'objet.

Vous pouvez également afficher, télécharger et supprimer les versions d'un objet dans le panneau de présentation de l'objet. Pour plus d'informations, consultez Affichage d'une présentation d'un objet dans la console Amazon S3.

> **Important**
>
> Vous pouvez annuler la suppression d'un objet uniquement si celui-ci a été supprimé en tant que version la plus récente (version actuelle). Vous ne pouvez pas restaurer une version précédente d'un objet supprimé. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Utilisation des AWS SDK

Les exemples de chargement d'objets dans des compartiments non versionnés et activés pour la gestion des versions sont les mêmes. Toutefois, pour les compartiments activés pour la gestion des versions, Simple Storage Service (Amazon S3) attribue un numéro de version. Sinon, le numéro de version est null.

Pour des exemples de téléchargement d'objets à l'aide de AWS kits SDK pour Java, .NET et PHP, consultez la section Téléchargement d'objets.

Pour des exemples de liste des versions d'objets à l'aide de AWS kits SDK pour .NET et Rust, consultez Répertorier la version des objets dans un compartiment Amazon S3.

Utilisation de l'API REST

Pour récupérer une version d'objet spécifique

1. Configurez la valeur `versionId` sur l'ID de la version de l'objet que vous souhaitez récupérer.
2. Envoyez une demande `GET Object versionId`.

Example Récupération d'un objet soumis à la gestion des versions

La demande suivante récupère la version `L4kqtJlcpXroDTDmpUMLUo` de `my-image.jpg`.

```
GET /my-image.jpg?versionId=L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Seules les métadonnées d'un objet peuvent être récupérées, et pas son contenu. Pour plus d'informations, consultez the section called “Récupération des métadonnées de version”.

Pour plus d'informations sur la restauration d'une version d'objet précédente, consultez the section called “Restauration des versions précédentes”.

Récupération des métadonnées d'une version d'objet

Si vous souhaitez uniquement récupérer les métadonnées d'un objet (et pas son contenu), vous utilisez l'opération `HEAD`. Par défaut, vous obtenez les métadonnées de la version la plus récente. Pour récupérer les métadonnées d'une version d'objet spécifique, vous spécifiez son ID de version.

Pour récupérer les métadonnées d'une version d'objet

1. Configurez la valeur `versionId` sur l'ID de la version de l'objet dont vous souhaitez récupérer les métadonnées.
2. Envoyez une demande `HEAD Object versionId`.

Example — Récupération des métadonnées d'un objet soumis à la gestion des versions

La requête suivante récupère les métadonnées de la version `3HL4kqCxf3vjVBH40Nrjfkd` de `my-image.jpg`.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Voici un exemple de réponse.

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Restauration des versions précédentes

Vous pouvez utiliser la gestion des versions pour récupérer les versions précédentes d'un objet. Pour ce faire, deux approches sont possibles :

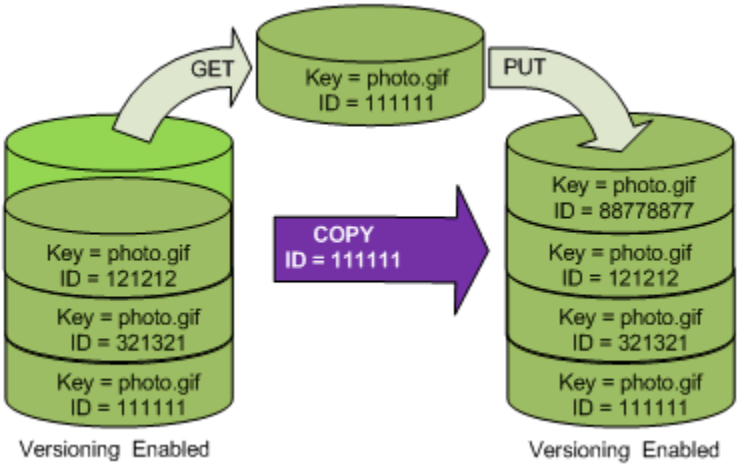
- La copie d'une version précédente de l'objet dans le même compartiment.

  L'objet copié devient la version actuelle de cet objet et toutes les versions d'objet sont préservées.
- La suppression définitive de la version actuelle de l'objet.

  Lorsque vous supprimez la version d'objet actuelle, vous transformez la version précédente en version actuelle de cet objet.

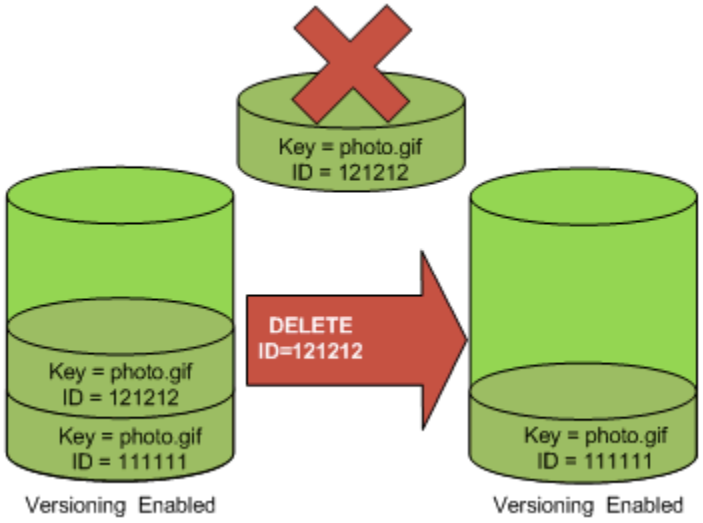
Etant donné que toutes les versions d'objet sont préservées, vous pouvez réaliser une version antérieure à la version actuelle en copiant une version spécifique de l'objet dans le même compartiment. Dans le schéma suivant, l'objet source (ID = 111111) est copié dans le même

compartiment. Amazon S3 fournit un nouvel ID (88778877) qui devient la version actuelle de l'objet. Le compartiment possède donc la version d'objet originale (111111) et sa copie (88778877). Pour plus d'informations sur l'obtention d'une version précédente, puis son chargement pour en faire la version actuelle, consultez Récupération des versions d'objets à partir d'un compartiment activé pour la gestion des versions et Chargement d'objets.



Une version ultérieure GET récupère la version 88778877.

Le schéma suivant illustre comment la suppression de la version actuelle (121212) d'un objet permet à la version précédente (111111) de devenir l'objet actuel. Pour plus d'informations sur la suppression d'un objet, consultez Suppression d'un objet unique.



Une version ultérieure GET récupère la version 111111.

> **Note**
>
> Pour restaurer des versions d'objets par lots, vous pouvez utiliser l'opération `CopyObject`. L'opération `CopyObject` copie chaque objet spécifié dans le manifeste. Cependant, sachez que les objets ne sont pas nécessairement copiés dans le même ordre que celui dans lequel ils apparaissent dans le manifeste. Pour les compartiments activés pour le contrôle de version, si la préservation de l'ordre des versions actuelles ou anciennes est importante, vous devez d'abord copier toutes les anciennes versions. Ensuite, une fois la première tâche terminée, copiez les versions actuelles dans une tâche ultérieure.

Pour restaurer les versions d'objets précédentes

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, choisissez le nom de l'objet.
4. Choisissez Versions.

   Amazon S3 affiche toutes les versions de l'objet.
5. Sélectionnez la case à cocher en regard de l'ID de version des versions que vous souhaitez récupérer.
6. Choisissez Actions, choisissez Téléchargeret enregistrez l'objet.

Vous pouvez également afficher, télécharger et supprimer les versions d'un objet dans le panneau de présentation de l'objet. Pour plus d'informations, consultez Affichage d'une présentation d'un objet dans la console Amazon S3.

> **Important**
>
> Vous pouvez annuler la suppression d'un objet uniquement si celui-ci a été supprimé en tant que version la plus récente (version actuelle). Vous ne pouvez pas restaurer une version précédente d'un objet supprimé. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Utilisation des AWS SDK

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez le AWS Developer Center.

Python

L'exemple de code Python suivant restaure la version précédente d'un objet versionné en supprimant toutes les versions qui sont apparues après la version de restauration spécifiée.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
```

```
                print(f"Deleted version {version.version_id}")
            else:
                break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
f"{object_key}."
        )
```

Suppression des versions d'objet d'un compartiment activé pour la gestion des versions

Vous pouvez supprimer des versions d'objets des compartiments Amazon S3 quand vous le souhaitez. Vous pouvez également définir des règles de configuration du cycle de vie pour les objets qui possèdent un cycle de vie bien défini pour demander à Amazon S3 d'expirer les versions d'objet actuelles ou de supprimer définitivement les anciennes versions d'objet. Lorsque le compartiment est activé ou désactivé pour la gestion des versions, les actions de la configuration du cycle de vie fonctionnent comme suit :

- L'action `Expiration` s'applique à la version actuelle de l'objet. Au lieu de supprimer la version actuelle de l'objet, Amazon S3 la conserve en tant que version ancienne en ajoutant un marqueur de suppression, qui devient ensuite la version actuelle.
- L'action `NoncurrentVersionExpiration` s'applique aux anciennes versions d'objet, et Amazon S3 les supprime définitivement. Vous ne pouvez pas récupérer définitivement les objets supprimés.

Pour plus d'informations sur le cycle de vie S3, consultez Gestion du cycle de vie de votre stockage et Exemples de configuration de cycle de vie S3

Pour connaître le nombre de versions actuelles et anciennes des objets de vos compartiments, vous pouvez utiliser les métriques Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Using S3 Storage Lens to optimize your storage costs (Utilisation de S3 Storage Lens pour optimiser

vos coûts de stockage). Pour obtenir la liste complète des métriques, consultez le Glossaire des métriques S3 Storage Lens.

> **Note**
>
> Les tarifs habituels d'Amazon S3 s'appliquent à chaque version d'un objet stockée et transférée, y compris les versions d'objet non actuelles. Pour plus d'informations, consultez Tarification Amazon S3.

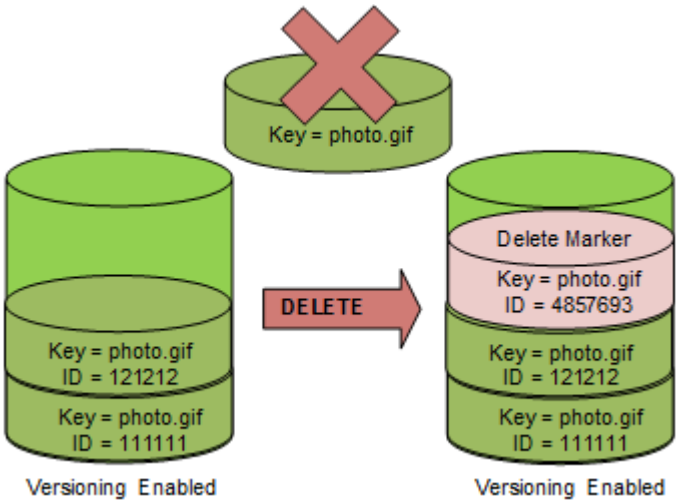
Supprimer les cas d'utilisation de demandes

Une demande `DELETE` possède les cas d'utilisation suivants :

- Lorsque la gestion des versions est activée, une simple demande `DELETE` ne peut pas supprimer définitivement un objet. (Une demande `DELETE` simple est une demande qui ne spécifie pas d'ID de version.) À la place, Amazon S3 insère un marqueur de suppression dans le compartiment qui devient la version actuelle de l'objet avec un nouvel ID.

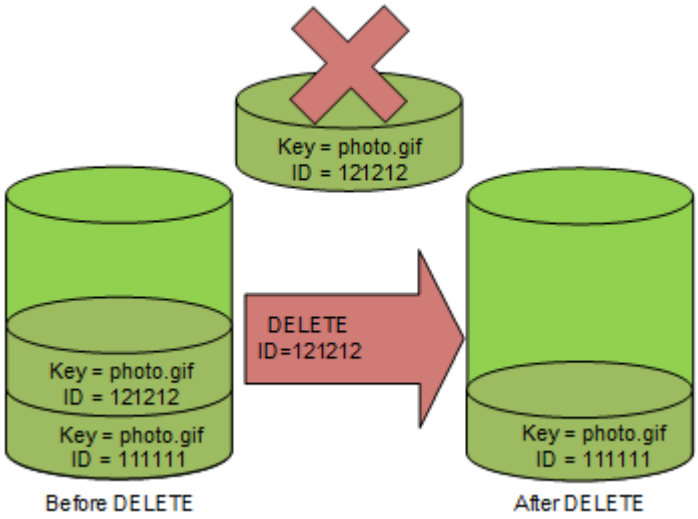
  Lorsque vous essayez de faire une demande `GET` sur un objet dont la version actuelle est un marqueur de suppression, Amazon S3 se comporte comme si l'objet avait été supprimé (même si ce n'est pas le cas) et renvoie une erreur 404. Pour plus d'informations, consultez Utilisation des marqueurs de suppression.

  Le schéma suivant montre qu'une simple demande `DELETE` ne supprime pas réellement l'objet spécifié. Au lieu de cela, Amazon S3 insère un marqueur de suppression.



- Pour supprimer définitivement les objets soumis à la gestion des versions, vous devez utiliser `DELETE Object versionId`.

  Le schéma suivant montre que la suppression d'un objet spécifié supprime définitivement cet objet.



Pour supprimer des versions d'un objet

Vous pouvez supprimer des versions d'objets dans Amazon S3 à l'aide de la console, AWS des SDK, de l'API REST ou du AWS Command Line Interface.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
3. Dans la liste Objets, choisissez le nom de l'objet.
4. Choisissez Versions.

   Amazon S3 affiche toutes les versions de l'objet.
5. Sélectionnez la case à cocher en regard de l'ID de version des versions que vous supprimer définitivement.
6. Choisissez Supprimer.
7. Dans Supprimer définitivement les objets ? , saisissez **`permanently delete`**.

> **Warning**
>
> Lorsque vous supprimez définitivement une version d'objet, l'action ne peut pas être annulée.

8. Choisissez Delete objects (Supprimer les objets).

   Amazon S3 supprime la version de l'objet.

Utilisation des AWS SDK

Pour des exemples de suppression d'objets à l'aide AWS des SDK pour Java, .NET et PHP, consultezSuppression d'objets Amazon S3. Les exemples de suppression d'objets dans des compartiments non versionnés et activés pour la gestion des versions sont les mêmes. Toutefois, pour les compartiments activés pour la gestion des versions, Simple Storage Service (Amazon S3) attribue un numéro de version. Sinon, le numéro de version est null.

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez le AWS Developer Center.

Python

L'exemple de code Python suivant supprime définitivement un objet assorti d'une gestion des versions en supprimant toutes ses versions.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Utilisation de l'API REST

Pour supprimer une version spécifique d'un objet

- Dans une demande `DELETE`, spécifiez un ID de version.

Example — Suppression d'une version spécifique

L'exemple suivant supprime la version `UIORUnfnd89493jJFJ` de `photo.gif`.

```
DELETE /photo.gif?versionId=UIORUnfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

En utilisant le AWS CLI

La commande suivante supprime un objet nommé test.txt d'un compartiment nommé *`example-s3-bucket1`*. Pour supprimer une version spécifique d'un objet, vous devez être le propriétaire du compartiment et vous devez utiliser la sous-ressource d'ID de version.

```
aws s3api delete-object --bucket example-s3-bucket1 --key test.txt --version-
id versionID
```

Pour plus d'informations sur `delete-object`, consultez delete-object dans la référence des commandes AWS CLI .

Pour plus d'informations sur la suppression des versions d'objet, veuillez consulter ces rubriques :

- Utilisation des marqueurs de suppression
- Suppression des marqueurs de suppression pour rendre une version plus ancienne à jour
- Suppression d'un objet à partir d'un compartiment activé pour la fonction Supprimer MFA

Utilisation des marqueurs de suppression

Dans Amazon S3, un marqueur de suppression désigne un espace réservé (ou marqueur) pour un objet soumis à la gestion des versions qui a été spécifié dans une demande `DELETE` simple. Une demande `DELETE` simple est une demande qui ne spécifie pas d'ID de version. Étant donné que l'objet était dans un compartiment activé pour la gestion des versions, il n'a pas été supprimé. Toutefois, à cause du marqueur de suppression, Amazon S3 se comporte comme si l'objet est supprimé. Vous pouvez utiliser un appel `DELETE` d'API Amazon S3 sur un marqueur de suppression. Pour ce faire, vous devez effectuer la `DELETE` demande en utilisant un utilisateur ou un rôle AWS Identity and Access Management (IAM) doté des autorisations appropriées.

Un marqueur de suppression possède un nom de clé (ou clé) et un ID de version comme tout autre objet. Toutefois, un marqueur de suppression diffère des autres objets des manières suivantes :

- Un marqueur de suppression n'a pas de données qui lui sont associées.
- Un marqueur de suppression n'est associé à aucune valeur de liste de contrôle d'accès (ACL).
- Si vous émettez une demande `GET` pour un marqueur de suppression, la demande `GET` ne récupère rien, car le marqueur de suppression ne contient aucune donnée. Plus précisément, lorsque votre demande `GET` ne spécifie pas de `versionId`, vous obtenez une erreur 404 (Introuvable).

Les marqueurs de suppression accumulent une charge minimale pour le stockage dans Amazon S3. La taille de stockage d'un marqueur de suppression est égale à la taille du nom de clé du marqueur de suppression. Un nom de clé est une séquence de caractères Unicode. L'encodage UTF-8 du nom de clé ajoute 1 à 4 octets de stockage à votre compartiment pour chaque caractère dans le nom. Les marqueurs de suppression sont stockés dans la classe de stockage S3 standard.

Si vous souhaitez connaître le nombre de marqueurs de suppression dont vous disposez et la classe de stockage dans laquelle ils sont stockés, vous pouvez utiliser Amazon S3 Storage Lens. Pour plus d’informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec Amazon S3 Storage Lens et Glossaire des métriques Amazon S3 Storage Lens.

Pour en savoir plus sur les noms de clé, consultez Création de noms de clés d'objet. Pour plus d'informations sur le marqueur de suppression, consultez Gestion des marqueurs de suppression.

Seul Simple Storage Service (Amazon S3) peut créer un marqueur de suppression, et il le fait dès que vous envoyez une demande `DeleteObject` sur un objet dans un compartiment activé ou désactivé pour la gestion des versions. L'objet spécifié dans la demande `DELETE` n'est pas

réellement supprimé. A la place, le marqueur de suppression devient la version actuelle de l'objet. Le nom de clé de l'objet (ou clé) devient la clé du marqueur de suppression.

Quand vous obtenez un objet sans spécifier de `versionId` dans votre demande, si sa version actuelle est un marqueur de suppression, Amazon S3 fournit les réponses suivantes :

- Une erreur 404 (Introuvable)
- En-tête de réponse, `x-amz-delete-marker: true`

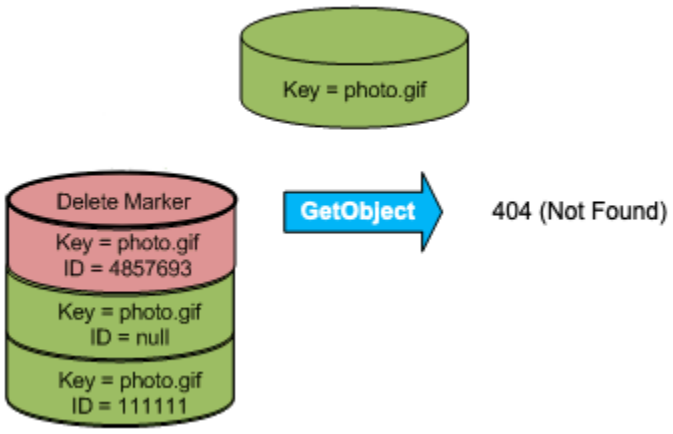
Quand vous obtenez un objet en spécifiant un `versionId` dans votre demande, si la version actuelle est un marqueur de suppression, Amazon S3 fournit les réponses suivantes :

- Une erreur 405 (méthode non autorisée)
- En-tête de réponse, `x-amz-delete-marker: true`
- Un en-tête de réponse `Last-Modified: timestamp` (uniquement lors de l'utilisation des opérations HeadObjectou de GetObjectl'API)

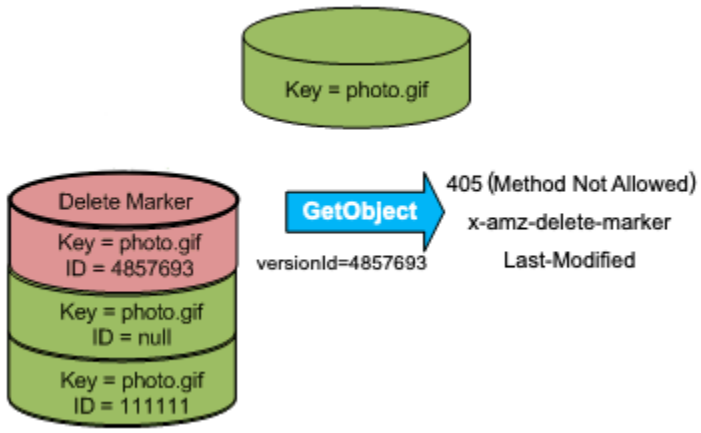
L'en-tête de réponse `x-amz-delete-marker: true` vous indique que l'objet consulté était un marqueur de suppression. Cet en-tête de réponse ne renvoie jamais `false`, car lorsque la valeur est `false`, la version actuelle ou spécifiée de l'objet n'est pas un marqueur de suppression.

L'en-tête de réponse `Last-Modified` indique l'heure de création des marqueurs de suppression.

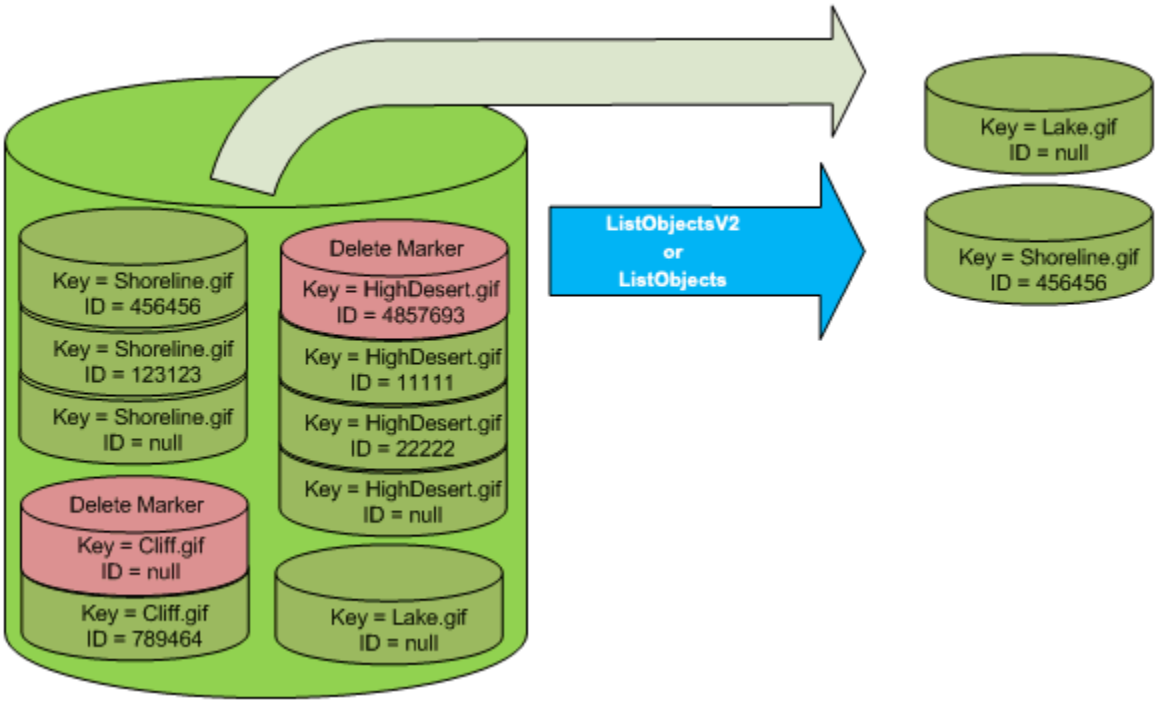
La figure suivante montre comment un appel d'API `GetObject` sur un objet dont la version actuelle est un marqueur de suppression répond avec une erreur 404 (Introuvable) et avec un en-tête de réponse incluant `x-amz-delete-marker: true`.



Si vous effectuez un appel `GetObject` sur un objet en spécifiant un `versionId` dans votre demande, et si la version spécifiée est un marqueur de suppression, Amazon S3 répond avec une erreur 405 (Méthode non autorisée) et les en-têtes de réponse incluent `x-amz-delete-marker: true` et `Last-Modified: timestamp`.



Le seul moyen de lister des marqueurs de suppression (et d'autres versions d'un objet) est d'utiliser la sous-ressource `versions` dans une demande ListObjectVersions. La figure suivante montre qu'une demande ListObjectsV2 ou ListObjects ne renvoie pas d'objets dont la version actuelle est un marqueur de suppression.



Gestion des marqueurs de suppression

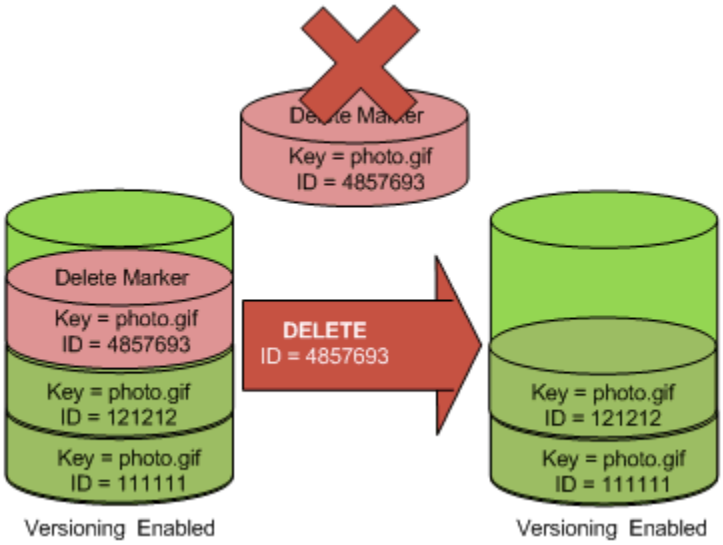
Configuration du cycle de vie pour nettoyer automatiquement les marqueurs de suppression expirés

Un marqueur de suppression d'objet expiré est un marqueur dans lequel toutes les versions d'objet sont supprimées et où il ne reste qu'un seul marqueur de suppression. Si la configuration de cycle de vie est définie pour supprimer les versions actuelles, ou si l'action `ExpiredObjectDeleteMarker` est explicitement définie, Amazon S3 supprime le marqueur de suppression de l'objet expiré. Pour voir un exemple, consultez Exemple 7 : Suppression des marqueurs de suppression d'objet expiré.

Suppression des marqueurs de suppression pour rendre une version plus ancienne à jour

Lorsque vous supprimez un objet d'un compartiment activé pour la gestion des versions, toutes les versions restent dans le compartiment et Simple Storage Service (Amazon S3) crée un marqueur de suppression pour l'objet. Pour annuler la suppression de l'objet, vous devez supprimer ce marqueur de suppression. Pour plus d'informations sur la gestion des versions et les marqueurs de suppression, consultez Utilisation de la gestion des versions dans les compartiments S3.

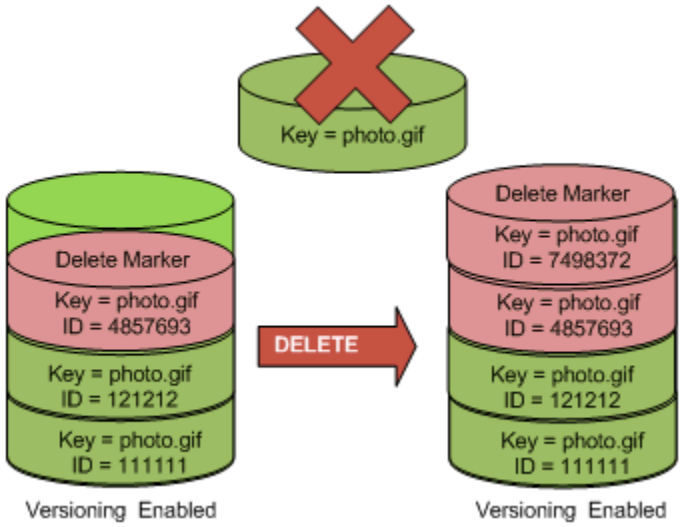
Pour supprimer un marqueur de suppression définitivement, vous devez inclure son ID de version dans une demande `DeleteObject versionId`. Le schéma suivant montre comment une demande `DeleteObject versionId` supprime définitivement un marqueur de suppression.



Suite à la suppression du marqueur de suppression, une simple demande `GET` récupère désormais l'ID de version actuelle (121212) de l'objet.

> **Note**
>
> Si vous utilisez une demande `DeleteObject` où la version actuelle est un marqueur de suppression (sans spécifier l'ID de version du marqueur de suppression), Amazon S3 ne supprime pas le marqueur de suppression, mais à la place, `PUT`s un autre marqueur de suppression.

Pour supprimer un marqueur de suppression avec un ID de version `NULL`, vous devez transmettre l'ID de version `NULL` comme ID de version dans la demande `DeleteObject`. La figure suivante illustre comment un demande `DeleteObject` adressée sans ID de version, où la version actuelle est un marqueur de suppression, ne supprime rien, mais ajoute à la place un marqueur de suppression supplémentaire avec un ID de version unique (7498372).



Utilisation de la console S3

Suivez ces étapes pour récupérer des objets supprimés qui ne sont pas des dossiers de votre compartiment S3, y compris les objets qui se trouvent dans ces dossiers.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez.
3. Pour afficher la liste des versions des objets du compartiment, choisissez le commutateur Répertorier les versions. Vous pourrez voir les marqueurs de suppression pour les objets supprimés.

4. Pour annuler la suppression d'un objet, vous devez supprimer le marqueur de suppression. Cochez la case en regard du marqueur de suppression de l'objet à récupérer, puis choisissez Delete (Supprimer).
5. Confirmez la suppression sur la page Delete objects (Supprimer les objets).

   a. Pour Pemanently delete objects? (Supprimer définitivement des objets ?), saisissez **`permanently delete`**.

   b. Choisissez Delete objects (Supprimer les objets).

> **Note**
>
> Vous ne pouvez pas utiliser la console Amazon S3 pour restaurer des dossiers. Vous devez utiliser le AWS CLI ou le SDK. Pour des exemples, veuillez consulter Comment récupérer un objet Simple Storage Service (Amazon S3) supprimé d'un compartiment activé pour la gestion des versions ? dans le Centre de connaissances AWS .

Utilisation de l'API REST

Pour supprimer définitivement un marqueur de suppression

1. Configurez la valeur `versionId` sur l'ID de la version du marqueur de suppression que vous souhaitez supprimer.
2. Envoyez une demande `DELETE Object versionId`.

Example — Suppression d'un marqueur de suppression

L'exemple suivant supprime le marqueur de suppression pour la version 4857693 `photo.gif`.

```
DELETE /photo.gif?versionId=4857693 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Lorsque vous supprimez un marqueur de suppression, Amazon S3 inclut les éléments suivants dans la réponse.

```
204 NoContent
```

```
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Utilisation des AWS SDK

Pour plus d'informations sur l'utilisation d'autres AWS SDK, consultez le AWS Developer Center.

Python

L'exemple de code Python suivant montre comment supprimer un marqueur de suppression d'un objet et transformer ainsi la version ancienne la plus récente en la version actuelle de l'objet.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
 marker.
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
                latest_version["LastModified"],
            )
            obj = bucket.Object(object_key)
            obj.Version(latest_version["VersionId"]).delete()
            logger.info(
```

```
                "Revived %s, active version is now %s  with body '%s'",
                object_key,
                obj.version_id,
                obj.get()["Body"].read(),
            )
        else:
            logger.warning(
                "Delete marker is not the latest version for %s!", object_key
            )
    elif "Versions" in response:
        logger.warning("Got an active version for %s, nothing to do.", object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Suppression d'un objet à partir d'un compartiment activé pour la fonction Supprimer MFA

Si la fonction Supprimer MFA est activée dans la configuration de la gestion des versions d'un compartiment, le propriétaire du compartiment doit inclure l'en-tête de la demande `x-amz-mfa` dans les demandes pour supprimer définitivement une version d'objet ou modifier l'état de la gestion des versions du compartiment. Les demandes qui incluent `x-amz-mfa` doivent utiliser HTTPS.

La valeur de l'en-tête est la concaténation du numéro de série du périphérique d'authentification, un espace et le code d'authentification affiché dessus. Si vous n'incluez pas cet en-tête de la demande, cette dernière échoue.

Pour en savoir plus sur les périphériques d'authentification, veuillez consulter la section Authentification multifacteur.

Example — Suppression d'un objet à partir d'un compartiment activé pour la fonction Supprimer MFA

L'exemple suivant montre comment supprimer `my-image.jpg` (avec la version spécifiée), qui se trouve dans un compartiment où la fonction Suppression MFA est activée.

Notez l'espace entre *`[SerialNumber]`* et *`[AuthenticationCode]`*. Pour plus d'informations, veuillez consulter DeleteObject dans la Référence d'API Amazon Simple Storage Service.

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
```

```
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Pour en savoir plus sur l'activation de la fonction Supprimer MFA, consultez Configuration de la fonction Supprimer MFA.

Configuration des autorisations d'objet soumis à la gestion des versions

Les autorisations pour les objets dans Amazon S3 sont définies au niveau de la version. Chaque version est dotée de son propre propriétaire d'objet. Celui Compte AWS qui crée la version de l'objet est le propriétaire. Vous pouvez donc configurer différentes autorisations pour différentes versions du même objet. Pour ce faire, vous devez spécifier l'ID de version de l'objet dont vous souhaitez configurer les autorisations dans une demande `PUT Object versionId acl`. Pour une description détaillée et des instructions sur l'utilisation des listes ACL, consultez Identity and Access Management pour Amazon S3.

Example — Configuration des autorisations pour une version d'objet

La requête suivante définit les autorisations du bénéficiaire, `BucketOwner@amazon.com`, à `FULL_CONTROL` sur la clé, `my-image.jpg`, ID de version, `3HL4kqtJvjVBH40Nrjfkd`.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
```

```
  </AccessControlPolicy>
```

De la même façon, pour obtenir les autorisations d'une version d'objet spécifique, vous devez spécifier son ID de version dans une demande `GET Object versionId acl`. Vous devez inclure l'ID de version car, par défaut, la demande `GET Object acl` renvoie les autorisations de la version actuelle de l'objet.

Example — Récupération des autorisations pour une version d'objet spécifique

Dans l'exemple suivant, Amazon S3 renvoie les autorisations pour la clé, `my-image.jpg`, l'ID de version, `DVBH40Nr8X8gUMLUo`.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Pour plus d'informations, veuillez consulter GetObjectAcl dans la Référence d'API Amazon Simple Storage Service.

Utilisation des objets dans un compartiment désactivé pour la gestion des versions

Dans Simple Storage Service (Amazon S3), vous pouvez désactiver la gestion des versions pour stopper l'accumulation de nouvelles versions du même objet dans un compartiment. Vous pouvez le faire car vous ne souhaitez qu'une seule version d'un objet dans un compartiment. Ou, vous ne voudrez peut-être pas accumuler de frais pour plusieurs versions.

Lorsque vous désactivez la gestion des versions, les objets existants du compartiment ne changent pas. Seule la façon dont Amazon S3 gère les objets dans les futures demandes change. Les rubriques de cette section expliquent les diverses opérations d'objet dans un compartiment dont la gestion des versions est désactivée, notamment l'ajout, la récupération et la suppression des objets.

Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3. Pour plus d'informations sur la récupération des versions d'objet, consultez Récupération de versions d'objets à partir d'un compartiment activé pour la gestion des versions.

Rubriques

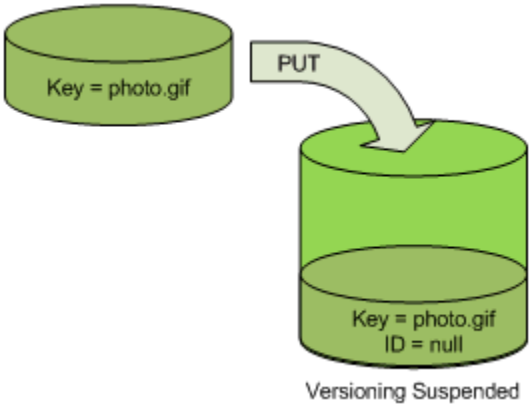
- Ajout d'objets dans des compartiments désactivés pour la gestion des versions
- Récupération d'objets à partir des compartiments désactivés pour la gestion des versions
- Suppression d'objets à partir des compartiments désactivés pour la gestion des versions

Ajout d'objets dans des compartiments désactivés pour la gestion des versions

Vous pouvez ajouter des objets aux compartiments désactivés pour la gestion des versions dans Simple Storage Service (Amazon S3) afin de créer l'objet avec un ID de version null ou bien pour écraser toute version d'objet avec un ID de version correspondant.

Après avoir désactivé la gestion des versions sur un compartiment, Simple Storage Service (Amazon S3) ajoute ensuite automatiquement un ID de version `null` à chaque objet stocké ultérieurement (à l'aide de la demande `PUT`, `POST`, ou `CopyObject`) dans le compartiment.

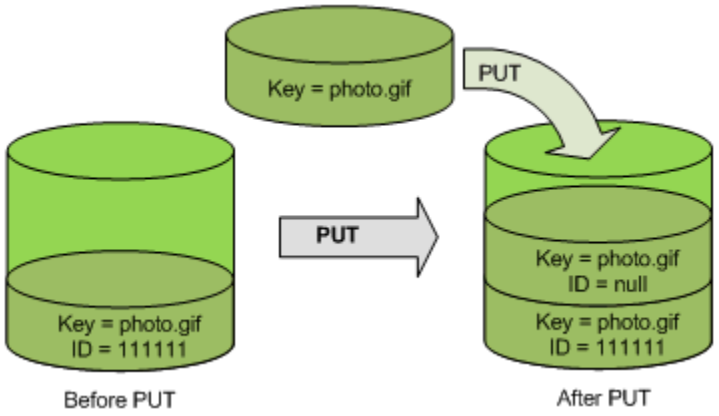
Le schéma suivant montre comment Amazon S3 ajoute l'ID de version `null` à un objet lorsqu'il est ajouté à un compartiment désactivé pour la gestion des versions.



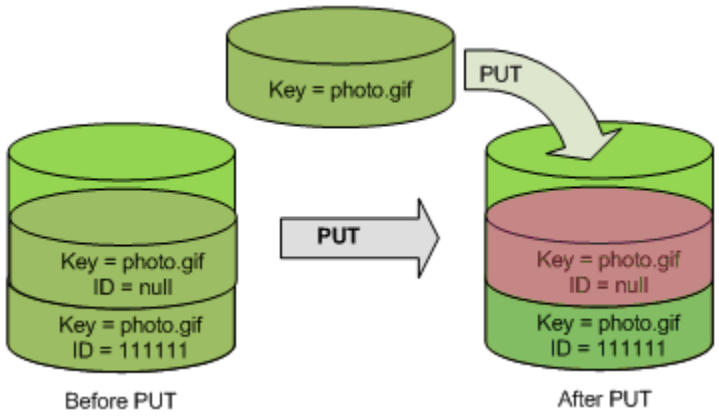
Si une version null est déjà dans le compartiment et que vous ajoutez un autre objet avec la même clé, l'objet ajouté remplace la version nulle originale.

En cas de présence d'objets versionnés dans le compartiment, la version pour laquelle vous faites une demande PUT devient la version actuelle de l'objet. Le schéma suivant montre en quoi l'ajout d'un objet à un compartiment qui contient des objets versionnés ne remplace pas l'objet déjà dans le compartiment.

Dans ce cas, la version 111111 était déjà dans le compartiment. Amazon S3 attache un ID de version null à l'objet en train d'être ajouté et le stocke dans le compartiment. La version 111111 n'est pas remplacée.



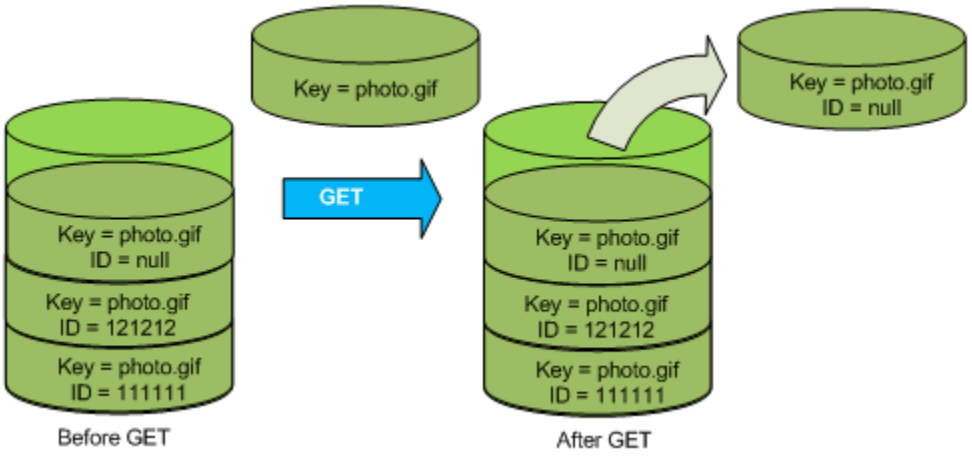
Si une version null existe déjà dans un compartiment, elle est remplacée, comme illustré dans le schéma suivant.



Notez que même si la clé et l'ID de version (`null`) de la version null sont identiques avant et après la demande PUT, le contenu de la version null initialement stocké dans le compartiment est remplacé par le contenu de l'objet PUT dans le compartiment.

Récupération d'objets à partir des compartiments désactivés pour la gestion des versions

Une demande `GET Object` renvoie la version actuelle d'un objet que vous ayez activé ou non la gestion des versions sur un compartiment. Le schéma suivant montre comment une simple demande GET renvoie la version actuelle d'un objet.



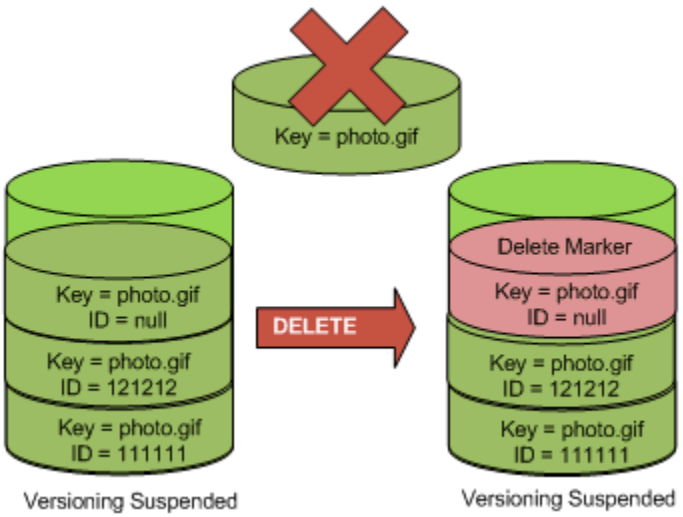
Suppression d'objets à partir des compartiments désactivés pour la gestion des versions

Vous pouvez supprimer des objets à partir des compartiments désactivés pour la gestion des versions afin de supprimer un objet ayant un ID de version null.

Si la gestion des versions est suspendue pour un compartiment, une demande `DELETE` :

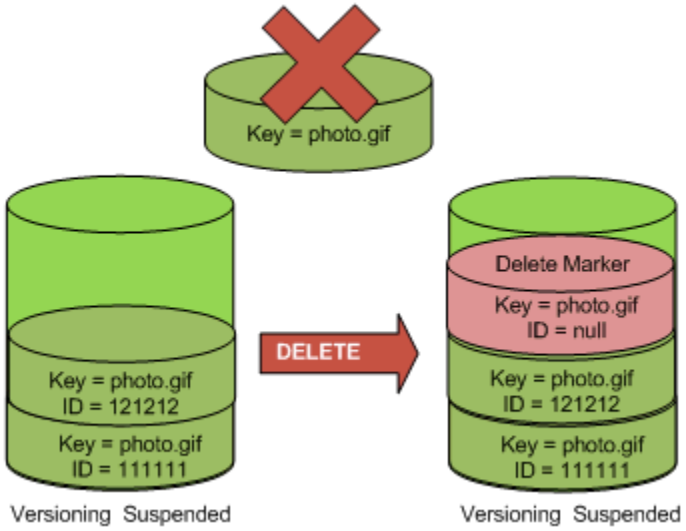
- Peut uniquement supprimer un objet dont l'ID de version est `null`.
- Ne supprime rien s'il n'y a aucune version null de l'objet dans le compartiment.
- Insère un marqueur de suppression dans le compartiment.

La figure suivante illustre comment un simple `DELETE` supprime une version null. (Une demande `DELETE` simple est une demande qui ne spécifie pas d'ID de version.) Amazon S3 insère un marqueur de suppression à sa place avec un ID de version de `null`.

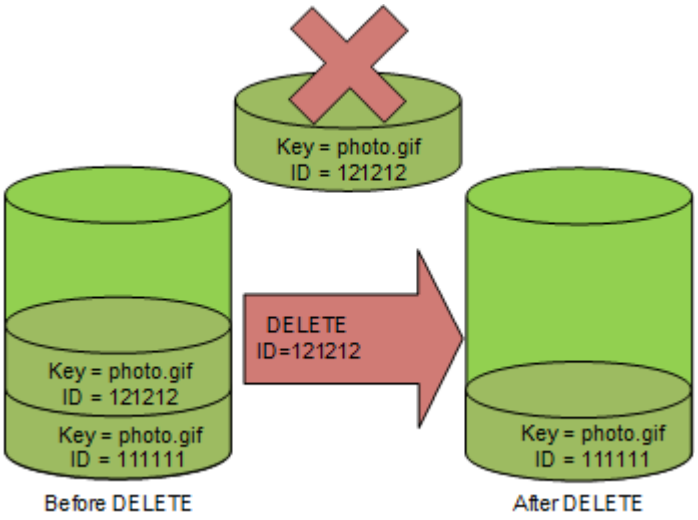


Rappelez-vous qu'un marqueur de suppression ne possède pas de contenu, donc vous perdez le contenu de la version null lorsqu'un marqueur de suppression la remplace.

Le schéma suivant montre un compartiment qui ne possède pas de version null. Dans ce cas, la demande DELETE ne supprime rien ; Simple Storage Service (Amazon S3) insère simplement un marqueur de suppression.



Même dans un compartiment dont la gestion des versions est suspendue, le propriétaire du compartiment peut supprimer définitivement une version spécifiée en incluant l'ID de la version dans la demande DELETE. Le schéma suivant montre que la suppression d'un objet spécifié supprime définitivement cette version de l'objet. Seul le propriétaire du compartiment peut supprimer une version d'objet spécifiée.



Utiliser AWS Backup pour Amazon S3

Amazon S3 est intégré nativement à AWS Backup, un service entièrement géré et basé sur des politiques que vous pouvez utiliser pour définir de manière centralisée des politiques de sauvegarde afin de protéger vos données dans Amazon S3. Après avoir défini vos politiques de sauvegarde et affecté des ressources Amazon S3 aux politiques, AWS Backup automatise la création de sauvegardes Amazon S3 et stocke en toute sécurité les sauvegardes dans un coffre de sauvegarde chiffré que vous spécifiez dans votre plan de sauvegarde.

Lorsque vous utilisez AWS Backup pour Amazon S3, vous pouvez effectuer les actions suivantes :

- Créez des sauvegardes continues et des sauvegardes périodiques. Les sauvegardes continues sont utiles pour une restauration à un instant donné dans le passé,et les sauvegardes périodiques sont utiles pour répondre à vos besoins de rétention de données à long terme.
- Automatisez la planification et la rétention des sauvegardes en configurant les stratégies de sauvegarde de manière centralisée.
- Restaurez les sauvegardes des données Amazon S3 à un point dans le temps que vous spécifiez.

Avec AWS Backup, vous pouvez utiliser la gestion des versions S3 et la réplication S3 afin de pouvoir mieux récupérer après des suppressions accidentelles et effectuer vos propres opérations d'auto-récupération.

Prérequis

Vous devez activer la gestion des versions S3 sur votre compartiment avant qu'AWS Backup ne puisse le sauvegarder.

> **Note**
>
> Nous vous recommandons de définir une règle d'expiration du cycle de vie pour les compartiments compatibles avec la gestion des versions qui sont sauvegardés. Si vous ne définissez pas de période d'expiration du cycle de vie, vos coûts de stockage Amazon S3 risquent d'augmenter car AWS Backup conserve toutes les versions de vos données Amazon S3.

Démarrer

Pour commencer à utiliser AWS Backup pour Amazon S3, consultez Création des sauvegardes Amazon S3 dans le Guide du développeur AWS Backup.

Restrictions et limitations

Pour en savoir plus sur les limites, consultez Création des sauvegardes Amazon S3 dans le Guide du développeur AWS Backup.

Utilisation des objets archivés

Pour réduire vos coûts de stockage pour les objets rarement consultés, vous pouvez archiver ces objets. Lorsque vous archivez un objet, il est placé dans un espace de stockage à faible coût, ce qui signifie que vous ne pouvez pas y accéder en temps réel.

Bien que les objets archivés ne soient pas accessibles en temps réel, vous pouvez les restaurer en quelques minutes ou quelques heures, selon la classe de stockage. Vous pouvez restaurer un objet archivé à l'aide de la console Amazon S3, de S3 Batch Operations, de l'API REST, AWS des SDK et du AWS Command Line Interface (AWS CLI). Pour obtenir des instructions, veuillez consulter Restauration d'un objet archivé.

Les objets Amazon S3 figurant dans les classes ou niveaux de stockage suivants sont archivés et ne sont pas accessibles en temps réel :

- Classe de stockage S3 Glacier Flexible Retrieval
- Classe de stockage S3 Glacier Deep Archive
- Niveau Accès aux archives S3 Intelligent-Tiering.
- Niveau d'accès Deep Archive de S3 Intelligent-Tiering

Pour restaurer les objets archivés, vous devez effectuer les opérations suivantes :

- Pour les objets des classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, vous devez d'abord lancer une demande de restauration, puis attendre qu'une copie temporaire de l'objet soit disponible. Lorsqu'une copie temporaire de l'objet restauré est créée, la classe de stockage de l'objet reste la même. (Une demande d'opération d'API HeadObject ou GetObject renvoie S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive en tant que classe de stockage.)
- Pour les objets appartenant aux niveaux d'accès Archive S3 Intelligent-Tiering et Deep Archive S3 Intelligent-Tiering, vous devez lancer la demande de restauration et attendre que l'objet soit déplacé dans le niveau d'accès Frequent.

Pour en savoir plus sur la comparaison de toutes les classes de stockage Amazon S3, veuillez consulter Utilisation des classes de stockage Simple Storage Service (Amazon S3). Pour plus d'informations sur S3 Intelligent-Tiering, consultez the section called “Fonctionnement de S3 Intelligent-Tiering”.

Restauration d'objets depuis S3 Glacier

Lorsque vous utilisez S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, Amazon S3 restaure une copie temporaire de l'objet uniquement pour la durée spécifiée. Une fois ce temps écoulé, il supprime la copie de l'objet restaurée. Vous pouvez modifier la période d'expiration d'une copie restaurée en réémettant une demande de restauration. Dans ce cas, Amazon S3 met à jour la période d'expiration relative à l'heure actuelle.

> **Note**
>
> Lorsque vous restaurez un objet archivé depuis S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, vous payez à la fois pour l'objet archivé et pour la copie que vous avez restaurée temporairement. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3.

Restauration des objets depuis S3 Intelligent-Tiering

Lorsque vous restaurez un objet à partir du niveau Archive Access de S3 Intelligent-Tiering ou du niveau Deep Archive Access de S3 Intelligent-Tiering, l'objet revient au niveau d'accès fréquent de S3 Intelligent-Tiering. Si vous n'accédez pas à l'objet dans 30 jours consécutifs, il est automatiquement déplacé vers le niveau Accès peu fréquent. Après un minimum de 90 jours consécutifs sans accès, l'objet passe au niveau Accès aux archives S3 Intelligent-Tiering. Si vous n'accédez pas à l'objet pendant un minimum de 180 jours consécutifs, il passe au niveau Accès Deep Archive.

> **Note**
>
> Contrairement aux classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive, les demandes de restauration pour les objets S3 Intelligent-Tiering n'acceptent pas la valeur `Days`.

Utilisation des opérations par lots S3 avec les demandes de restauration

Pour restaurer plusieurs objets Amazon S3 avec une seule demande, vous pouvez utiliser la fonctionnalité d'opérations par lots S3. Vous fournissez à la fonctionnalité d'opérations par lots S3 une liste d'objets sur lesquels effectuer des opérations. La fonctionnalité des opérations par lot S3 appelle l'opération d'API respective pour effectuer l'opération spécifiée. Une tâche d'opérations par lot peut effectuer l'opération spécifiée sur des milliards d'objets contenant des exaoctets de données.

Durée de restauration

Amazon S3 calcule le délai d'expiration de la copie d'objet restaurée en ajoutant le nombre de jours spécifié dans la demande de restauration au moment où la restauration demandée est terminée. Amazon S3 arrondit ensuite la date et l'heure obtenues au jour suivant à minuit heure UTC (Universal Coordinated Time). Par exemple, supposons qu'une copie d'objet restaurée a été créée le 15 octobre 2012 à 10h30 UTC et que la période de restauration a été spécifiée comme étant de trois jours. Dans ce cas, la copie restaurée expire le 19 octobre 2012 à minuit heure UTC, moment auquel Amazon S3 supprime la copie de l'objet.

Le temps nécessaire à l'exécution d'une tâche de restauration dépend de la classe de stockage d'archive ou du niveau de stockage que vous utilisez et de l'option de récupération que vous spécifiez : Expédié (uniquement disponible pour S3 Glacier Flexible Retrieval et le niveau Archive Access de S3 Intelligent-Tiering), Standard ou En bloc. Pour plus d’informations, consultez Options de récupération des archives.

Vous pouvez être averti à l'issue de la restauration grâce aux notifications d'événements Amazon S3. Pour plus d’informations, consultez Notifications d'événements Amazon S3.

Rubriques

- Options de récupération des archives
- Restauration d'un objet archivé

Options de récupération des archives

Voici les opérations relatives aux options de récupération lors de la restauration d'un objet archivé dans Amazon S3 :

- Expédié : accédez rapidement à vos données stockées dans la classe de stockage S3 Glacier Flexible Retrieval ou au niveau Archive Access de S3 Intelligent-Tiering. Vous pouvez utiliser

cette option lorsque des demandes urgentes occasionnelles concernant un sous-ensemble d'archives sont nécessaires. Pour tous les objets archivés, à l'exception des plus volumineux (plus de 250 Mo), les données auxquelles vous accédez à l'aide des récupérations rapides sont généralement disponibles en 1 à 5 minutes.

> **Note**
>
> Les récupérations accélérées sont une fonctionnalité premium et sont facturées au tarif des demandes et des récupérations accélérées.
> Pour obtenir des informations sur la tarification Amazon S3, consultez Tarification Amazon S3.

La capacité provisionnée aide à garantir que la capacité de récupération pour les récupérations rapides de S3 Glacier Flexible Retrieval est disponible lorsque vous en avez besoin. Pour plus d'informations, consultez Capacité provisionnée.

- Standard : accédez à tous vos objets archivés en quelques heures. Standard constitue l'option par défaut pour les demandes de récupération qui ne spécifient pas l'option de récupération. Les récupérations standard prennent généralement 3 à 5 heures pour les objets stockés dans la classe de stockage S3 Glacier Flexible Retrieval ou dans le niveau d'accès Archive de S3 Intelligent-Tiering. Ces récupérations se terminent généralement en 12 heures pour les objets stockés dans la classe de stockage S3 Glacier Deep Archive ou au niveau d'accès Deep Archive de S3 Intelligent-Tiering. Les récupérations standard sont gratuites pour les objets stockés dans S3 Intelligent-Tiering.

> **Note**
>
> - Pour les objets stockés dans la classe de stockage S3 Glacier Flexible Retrieval ou dans le niveau S3 Intelligent-Tiering Archive Access, les extractions standard initiées à l'aide de l'opération de restauration S3 Batch Operations commencent généralement en quelques minutes et se terminent en 3 à 5 heures.
> - Pour les objets appartenant à la classe de stockage S3 Glacier Deep Archive ou au niveau S3 Intelligent-Tiering Deep Archive Access, les extractions standard initiées à l'aide de l'opération de restauration Batch Operations commencent généralement dans les 9 heures et se terminent dans les 12 heures.

- En bloc : accédez à vos données avec l'option de récupération la moins coûteuse dans Amazon S3 Glacier. Avec les récupérations en bloc, vous pouvez récupérer de grandes quantités de données, même des pétaoctets, à moindre coût.

  Pour les objets stockés dans la classe de stockage S3 Glacier Flexible Retrieval ou dans le niveau S3 Intelligent-Tiering Archive Access, les extractions groupées se terminent généralement dans un délai de 5 à 12 heures. Pour les objets stockés dans la classe de stockage S3 Glacier Deep Archive ou dans le niveau S3 Intelligent-Tiering Deep Archive Access, ces extractions se terminent généralement dans les 48 heures.

  Les extractions en masse sont gratuites pour les objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Intelligent-Tiering.

Le tableau suivant récapitule les options de récupération d'archive. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3.

Pour effectuer une ou une `Bulk` extraction `ExpeditedStandard`, définissez l'élément de demande dans la `Tier` demande d'opération de l'RestoreObjectAPI REST sur l'option que vous souhaitez, ou sur l'équivalent dans le AWS Command Line Interface (AWS CLI) ou les AWS SDK. Si vous avez acheté une capacité provisionnée, toutes les récupérations rapides de type Expedited sont automatiquement transmises par le biais de votre capacité provisionnée.

Capacité provisionnée

La capacité provisionnée aide à garantir que votre capacité de récupération pour les récupérations accélérées depuis S3 Glacier Flexible Retrieval est disponible lorsque vous en avez besoin. Chaque unité de capacité garantit qu'au moins trois récupérations accélérées peuvent être effectuées toutes les 5 minutes et fournit jusqu'à 150 méga-octets par seconde (Mo/s) de débit de récupération.

Si votre charge de travail nécessite un accès extrêmement fiable et prévisible à un sous-ensemble de vos données en quelques minutes, envisagez d'acheter une capacité de récupération provisionnée. Sans capacité provisionnée, les récupérations accélérées peuvent ne pas être acceptées lors de périodes de demandes élevées. Si vous avez besoin d'un accès aux récupérations accélérées en toutes circonstances, nous vous recommandons d'acheter de la capacité de récupération provisionnée.

Les unités de capacité allouées sont allouées à un Compte AWS. Ainsi, le demandeur de la récupération accélérée des données doit acheter l'unité de capacité provisionnée, et non pas le propriétaire du compartiment.

Vous pouvez acheter de la capacité provisionnée à l'aide de la console Amazon S3, de la console Amazon S3 Glacier, de l'opération d'API REST Purchase Provisioned Capacity, AWS des SDK ou du. AWS CLI Pour de plus amples informations sur la tarification de la capacité provisionnée, veuillez consulter Tarification Amazon S3.

Taux de demandes d'initiation de restauration S3 Glacier

Lorsque vous lancez des demandes de restauration pour des objets stockés dans la classe de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, un quota de demandes de récupération est appliqué pour votre Compte AWS S3 Glacier prend en charge les demandes de restauration à un débit allant jusqu'à 1 000 transactions par seconde. Si ce taux est dépassé, des demandes valides sont limitées ou rejetées et Amazon S3 renvoie une erreur `ThrottlingException`.

En option, vous pouvez également utiliser les opérations par lots S3 pour récupérer un grand nombre d'objets stockés dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive avec une demande simple. Pour plus d'informations, consultez Exécution des opérations par lot à grande échelle sur des objets Amazon S3.

Restauration d'un objet archivé

Les objets Amazon S3 figurant dans les classes ou niveaux de stockage suivants sont archivés et ne sont pas accessibles en temps réel :

- Classe de stockage S3 Glacier Flexible Retrieval
- Classe de stockage S3 Glacier Deep Archive
- Niveau Accès aux archives S3 Intelligent-Tiering.
- Niveau d'accès Deep Archive de S3 Intelligent-Tiering

Les objets Amazon S3 stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive ne sont pas immédiatement accessibles. Pour accéder à un objet dans ces classes de stockage, vous devez en restaurer une copie temporaire de l'objet dans son compartiment S3 pendant une durée spécifiée (nombre de jours). Si vous souhaitez une copie permanente de l'objet, restaurez l'objet, puis créez-en une copie dans votre compartiment Amazon S3. La copie des objets restaurés n'est pas prise en charge dans la console Amazon S3. Pour ce type d'opération de copie, utilisez le AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST. Excepté dans le cas où vous effectuez une copie et modifiez sa classe de

stockage, l'objet sera toujours stocké dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Pour obtenir des informations sur l'utilisation de ces classes de stockage, consultez Classes de stockage pour les objets rarement consultés.

Pour accéder aux objets dans les niveaux Archive Access et Deep Archive Access de S3 Intelligent-Tiering, vous devez lancer une demande de restauration, puis attendre que l'objet soit placé au niveau d'accès fréquent. Lorsque vous restaurez un objet depuis les niveaux d'accès Archive et Deep Archive, l'objet retourne au niveau d'accès Fréquent. Pour obtenir des informations sur l'utilisation de ces classes de stockage, consultez Classe de stockage pour l'optimisation automatique des données avec des modèles d'accès inconnus ou irréguliers.

Pour obtenir des informations générales sur les objets archivés, consultez Utilisation des objets archivés.

> **Note**
>
> - Lorsque vous restaurez un objet archivé à partir des classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, vous payez à la fois pour l'objet archivé et pour la copie que vous avez restaurée temporairement.
> - Lorsque vous restaurez un objet depuis S3 Intelligent-Tiering, aucuns frais de récupération ne sont facturés pour les extractions standard ou en masse.
> - Les demandes de restauration ultérieures appelées sur des objets archivés qui ont déjà été restaurés sont facturées comme des `GET` demandes. Pour obtenir des informations sur la tarification, consultez Tarification Amazon S3.

Restauration d'un objet archivé

Vous pouvez restaurer un objet archivé à l'aide de la console Amazon S3, de l'API REST Amazon S3, AWS des SDK, du AWS Command Line Interface (AWS CLI) ou des opérations par lots S3.

Utilisation de la console S3

Restauration d'objets à l'aide de la console Amazon S3

Utilisez la procédure suivante pour restaurer un objet qui a été archivé dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, ou aux niveaux de stockage Archive Access ou Deep Archive Access de S3 Intelligent-Tiering.

Pour restaurer un objet archivé

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient les objets que vous souhaitez restaurer.
4. Dans la liste Objets, sélectionnez le ou les objets à restaurer, choisissez Actions, puis Lancer la restauration.
5. Si vous effectuez une restauration depuis S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, saisissez le nombre de jours pendant lesquels vous souhaitez que vos données archivées soient accessibles dans la zone Nombre de jours pendant lesquels la copie restaurée est disponible.
6. Dans Niveau d'extraction, effectuez l'une des opérations suivantes :

   - Choisissez Récupération en bloc ou Récupération standard, puis Commencer la restauration.
   - Choisissez Expedited retrieval (Récupération accélérée) (disponible uniquement pour S3 Glacier Flexible Retrieval ou l'accès Archive de S3 Intelligent-Tiering). Si vous restaurez un objet dans S3 Glacier Flexible Retrieval, vous pouvez choisir d'acheter une capacité provisionnée pour votre récupération accélérée. Si vous souhaitez acheter une capacité provisionnée, passez à l'étape suivante. Si ce n'est pas le cas, choisissez Commencer la restauration.

   > **Note**
   >
   > Les objets des niveaux Archive Access et Deep Archive Access de S3 Intelligent-Tiering sont restaurés automatiquement au niveau d'accès fréquent.

7. (Facultatif) Si vous restaurez un objet dans S3 Glacier Flexible Retrieval et que vous avez choisi Récupération accélérée, vous pouvez choisir d'acheter une capacité provisionnée. La capacité provisionnée n'est disponible que pour les objets se trouvant dans S3 Glacier Flexible Retrieval. Si vous disposez d'une capacité provisionnée, choisissez Commencer la restauration pour démarrer une récupération provisionnée.

Si vous disposez d'une capacité provisionnée, toutes vos récupérations accélérées sont effectuées avec votre capacité provisionnée. Pour plus d'informations, consultez Capacité provisionnée.

- Si vous ne disposez pas d'une capacité provisionnée et que vous ne voulez pas en acheter, choisissez Commencer la restauration.
- Si vous ne disposez pas d'une capacité provisionnée, mais que vous souhaitez acheter des unités de capacité provisionnée (PCU), choisissez Acheter des PCU. Dans la boîte de dialogue Acheter des PCU, choisissez le nombre de PCU que vous souhaitez acheter, confirmez votre achat, puis choisissez Acheter des PCU. Lorsque le message Réussite de l'achat s'affiche, choisissez Commencer la restauration pour démarrer la récupération provisionnée.

À l'aide du AWS CLI

Restauration d'objets depuis S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive

L'exemple suivant utilise la commande `restore-object` pour restaurer l'objet *dir1/example.obj* dans le compartiment *example-s3-bucket*pendant 25 jours.

```
aws s3api restore-object --bucket example-s3-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Si la syntaxe JSON utilisée dans cet exemple génère une erreur sur un client Windows, remplacez la demande de restauration par la syntaxe suivante :

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Restauration d'objets depuis le niveau Archive Access ou Deep Archive Access de S3 Intelligent-Tiering

L'exemple suivant utilise la commande `restore-object` pour restaurer l'objet *dir1/example.obj* dans le compartiment *example-s3-bucket* au niveau d'accès fréquent.

```
aws s3api restore-object --bucket example-s3-bucket --key dir1/example.obj --restore-request '{}'
```

> **Note**
>
> Contrairement aux classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive, les demandes de restauration pour les objets S3 Intelligent-Tiering n'acceptent pas la valeur `Days`.

Surveillance du statut de la restauration

Pour surveiller le statut de votre demande `restore-object`, utilisez la commande suivante `head-object` :

```
aws s3api head-object --bucket example-s3-bucket --key dir1/example.obj
```

Pour plus d'informations, consultez la section restore-object dans la référence des commandes AWS CLI .

Utilisation de l'API REST

Amazon S3 fournit une opération d'API pour vous permettre de lancer la restauration d'un objet archivé. Pour plus d'informations, veuillez consulter RestoreObject dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS SDK

Pour des exemples de restauration d'objets archivés dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive à l'aide des AWS kits de développement logiciel, consultez. Utilisation `RestoreObject` avec un AWS SDK ou une CLI

Utilisation des opérations par lot S3

Pour restaurer plusieurs objets archivés avec une seule demande, vous pouvez utiliser les opérations par lot S3. Vous fournissez à la fonctionnalité d'opérations par lot S3 une liste d'objets sur lesquels effectuer des opérations. La fonctionnalité des opérations par lot S3 appelle l'opération d'API respective pour effectuer l'opération spécifiée. Une tâche d'opérations par lot peut effectuer l'opération spécifiée sur des milliards d'objets contenant des exaoctets de données.

Pour créer une tâche d'opérations par lot, vous devez disposer d'un manifeste contenant uniquement les objets que vous souhaitez restaurer. Vous pouvez créer un manifeste à l'aide de S3 Inventory ou vous pouvez fournir un fichier CSV contenant les informations nécessaires. Pour plus d'informations, consultez the section called “Spécification d'un manifeste”.

Avant de créer et d'exécuter des tâches d'opérations par lot S3, vous devez accorder des autorisations à Amazon S3 pour effectuer des opérations par lot S3 en votre nom. Pour les autorisations requises, consultez the section called “Octroi d'autorisations”.

> **Note**
>
> Les tâches d'opérations par lot peuvent fonctionner sur les objets de classe de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive, ou sur les objets des niveaux de stockage Archive Access et Deep Archive Access de S3 Intelligent-Tiering. Les opérations par lot ne peuvent pas fonctionner sur les deux types d'objets archivés dans la même tâche. Pour restaurer des objets des deux types, vous devez créer des tâches Batch Operations. Pour plus d'informations sur l'utilisation des opérations par lot pour restaurer des objets archivés, consultez the section called “Restaurer des objets”.

Pour créer une tâche d'opérations par lot Lancer une restauration d'objet S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Choisissez Créer une tâche.
4. Pour Région AWS, choisissez la Région dans laquelle vous souhaitez créer votre tâche.
5. Sous Format du manifeste, choisissez le type de manifeste à utiliser.
   - Si vous choisissez Rapport d'inventaire S3, entrez le chemin d'accès de l'objet `manifest.json` généré par Amazon S3 dans le cadre du rapport d'inventaire au format CSV. Si vous souhaitez utiliser une version de manifeste autre que la plus récente, saisissez l'ID de version de l'objet `manifest.json`.
   - Si vous choisissez CSV, entrez le chemin vers un objet manifeste au format CSV. L'objet manifeste doit respecter le format décrit dans la console. Si vous souhaitez utiliser une version autre que la plus récente, vous pouvez éventuellement inclure l'ID de version de l'objet manifeste.
6. Choisissez Suivant.
7. Dans la section Opération, choisissez Restaurer.
8. Dans la section Restaurer, pour Restaurer la source, choisissez Glacier Flexible Retrieval ou Glacier Deep Archive ou Niveau Archive Access ou Deep Archive Access d'Intelligent-Tiering.

Si vous avez choisi Glacier Flexible Retrieval ou Glacier Deep Archive, entrez un nombre pour Nombre de jours pendant lesquels la copie restaurée est disponible.

Pour Niveau d'extraction, choisissez le niveau que vous souhaitez utiliser.

9. Choisissez Suivant.

10. Sur la page Configurer des options supplémentaires, remplissez les sections suivantes :

    - Dans la section Options supplémentaires, fournissez une description de la tâche et spécifiez un numéro de priorité pour la tâche. Un nombre plus élevé indique une priorité plus élevée. Pour plus d'informations, consultez the section called “Affectation d'une priorité de tâche”.
    - Dans la section Rapport d'achèvement, sélectionnez si les opérations par lot doivent créer un rapport d'achèvement. Pour plus d'informations sur les rapports d'achèvement, consultez the section called “Rapports de fin de tâche”.
    - Dans la section Autorisations, vous devez accorder des autorisations à Amazon S3 pour effectuer des opérations par lot en votre nom. Pour les autorisations requises, consultez the section called “Octroi d'autorisations”.
    - (Facultatif) Dans la section Balises de tâche, ajoutez des balises dans les paires clé-valeur. Pour plus d'informations, consultez the section called “Utilisation d'étiquettes”.

    Lorsque vous avez terminé, choisissez Suivant.

11. Sur la page Vérification, vérifiez les paramètres. Si vous devez apporter des modifications, choisissez Précédent. Sinon, choisissez Créer une tâche.

Pour plus d'informations sur les opérations par lot, consultez Restauration d'objets à l'aide d'opérations par lot et Création d'une tâche d'opérations par lot S3.

Vérification de l'état de restauration et de la date d'expiration

Vous pouvez vérifier l'état d'une demande de restauration ou sa date d'expiration à l'aide de la console Amazon S3, des notifications d'événements Amazon S3, de l' AWS CLI API REST Amazon S3 ou de l'API REST Amazon S3.

> **Note**
>
> Les objets restaurés à partir des classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont stockés uniquement pendant le nombre de jours que vous spécifiez. Les procédures suivantes renvoient la date d'expiration de ces copies.
> Les objets restaurés à partir des niveaux de stockage S3 Intelligent-Tiering Archive Access et Deep Archive Access n'ont pas de date d'expiration et sont replacés dans le niveau Frequent Access.

Utilisation de la console S3

Pour vérifier le statut de la restauration et la date d'expiration d'un objet dans la console Amazon S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment qui contient l'objet que vous restaurez.
4. Dans la liste Objets, sélectionnez l'objet que vous êtes en train de restaurer. La page de détails de l'objet s'affiche.
   - Si la restauration n'est pas terminée, en haut de la page, vous voyez une section qui stipule Restauration en cours.
   - Si la restauration est terminée, en haut de la page, vous voyez une section qui stipule Restauration terminée. Si vous effectuez une restauration depuis S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, cette section affiche également la Date d'expiration de la restauration. Amazon S3 supprimera à cette date la copie restaurée de votre objet archivé.

Utilisation des notifications d'événements Amazon S3

Vous pouvez être informé de la fin de la restauration d'un objet en utilisant l'`s3:ObjectRestore:Completed`action associée à la fonctionnalité de notifications d'événements Amazon S3. Pour plus d'informations sur l'activation des notifications d'événements, consultez Activation des notifications à l'aide d'Amazon SQS, Amazon SNS et. AWS Lambda Pour plus d'informations sur les différents types d'`ObjectRestore`événements, consultezthe section called “Types d'événements pris en charge pour SQS, SNS et Lambda”.

À l'aide du AWS CLI

Vérifiez l'état de restauration et la date d'expiration d'un objet à l'aide du AWS CLI

L'exemple suivant utilise la commande `head-object` pour afficher les métadonnées de l'objet *`dir1/example.obj`* dans le compartiment *`example-s3-bucket`*. Lorsque vous exécutez cette commande sur un objet en cours de restauration, Amazon S3 indique si la restauration est en cours et (le cas échéant) la date d'expiration.

```
aws s3api head-object --bucket example-s3-bucket --key dir1/example.obj
```

Résultat attendu (restauration en cours) :

```
{
    "Restore": "ongoing-request=\"true\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Résultat attendu (restauration terminée) :

```
{
    "Restore": "ongoing-request=\"false\", expiry-date=\"Wed, 12 Aug 2020 00:00:00 GMT
\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Pour plus d'informations à ce sujet`head-object`, consultez head-objectle manuel de référence des AWS CLI commandes.

Utilisation de l'API REST

Amazon S3 fournit une opération d'API qui vous permet de récupérer les métadonnées des objets. Pour vérifier le statut de restauration et la date d'expiration d'un objet archivé à l'aide de l'API REST, consultez HeadObject dans la Référence de l'API Amazon Simple Storage Service.

Mise à niveau de la vitesse d'une restauration en cours

Vous pouvez mettre à niveau la vitesse de la restauration pendant que cette dernière est en cours.

Pour mettre à niveau une restauration en cours vers un niveau plus rapide

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient les objets que vous souhaitez restaurer.
4. Dans la liste Objets, sélectionnez l'objet que vous êtes en train de restaurer. La page de détails de l'objet s'affiche. Sur la page de détails de l'objet, choisissez Mettre à niveau le niveau d'extraction. Pour de plus amples informations sur la vérification du statut de restauration d'un objet, veuillez consulter Vérification de l'état de restauration et de la date d'expiration.
5. Choisissez le niveau auquel vous souhaitez passer, puis choisissez Commencer la restauration.

Utilisation du verrouillage des objets S3

Le verrouillage d'objet S3 permet d'empêcher la suppression ou le remplacement d'objets Amazon S3 sur une période déterminée ou indéfinie. Object Lock utilise un modèle write-once-read-many(WORM) pour stocker des objets. Vous pouvez utiliser Object Lock pour répondre aux exigences réglementaires qui nécessitent le stockage WORM, ou pour ajouter un niveau de protection supplémentaire contre la modification ou la suppression d'objets.

> **Note**
>
> Le verrouillage d'objet S3 a été évalué par Cohasset Associates en vue de son utilisation dans les environnements soumis aux réglementations SEC 17a-4, CFTC et FINRA. Pour plus d'informations sur la conformité du verrouillage d'objet vis-à-vis de ces réglementations, consultez Cohasset Associates Compliance Assessment.

Le verrouillage des objets propose deux façons de gérer la rétention des objets : les périodes de rétention et les conservations à des fins juridiques. Une version d'un objet peut être assortie d'une période de rétention, d'une mise en suspens juridique ou des deux.

- Période de rétention : la période de rétention est une période fixe au cours de la laquelle un objet reste verrouillé. Vous pouvez définir une période de conservation unique pour des objets individuels. En outre, vous pouvez définir une période de rétention par défaut pour un compartiment S3. Vous pouvez également limiter les périodes de conservation minimale et maximale autorisées en utilisant la clé de `s3:object-lock-remaining-retention-days` condition figurant dans la politique de compartiment. Cela vous permet d'établir une plage de périodes de conservation et de limiter les périodes de conservation qui peuvent être plus ou moins longues que cette plage.
- Mise en suspens juridique : une mise en suspens juridique assure la même protection que la période de rétention, mais n'a pas de date d'expiration. La mise en suspens juridique reste en vigueur tant que vous ne la retirez pas explicitement. Les blocages légaux sont indépendants des périodes de conservation et sont placés sur des versions d'objets individuelles.

Le verrouillage d'objet fonctionne uniquement dans les compartiments pour lesquels la gestion des versions S3 est activée. Lorsque vous verrouillez une version d'objet, Amazon S3 stocke les informations de verrouillage dans les métadonnées de cette version d'objet. Le fait de définir une période de rétention ou une mise en suspens juridique sur un objet a pour effet de protéger uniquement la version spécifiée dans la demande. Les périodes de conservation et les blocages légaux n'empêchent pas la création de nouvelles versions de l'objet, ni ne suppriment les marqueurs à ajouter au-dessus de l'objet. Pour en savoir plus sur la gestion des versions S3, veuillez consulter Utilisation de la gestion des versions dans les compartiments S3.

Si vous placez un objet dans un compartiment qui contient déjà un objet protégé existant doté du même nom de clé d'objet, Amazon S3 crée une nouvelle version de cet objet. La version protégée existante de l'objet reste verrouillée conformément à sa configuration de rétention.

Fonctionnement du verrouillage d'objets S3

Rubriques

- Bonnes pratiques d'utilisation de S3 Object Lock
- Autorisations nécessaires

Périodes de rétention

Une période de rétention protège une version d'objet pendant une durée fixe. Lorsque vous mettez en place une période de rétention sur une version d'objet, Amazon S3 stocke un horodatage dans les métadonnées de la version d'objet pour indiquer la date à laquelle la période de rétention expire. Après l'expiration de la période de rétention, la version de l'objet peut être remplacée ou supprimée.

Vous pouvez définir une période de rétention de manière explicite sur une version d'objet individuelle ou sur les propriétés d'un compartiment afin qu'elle s'applique automatiquement à tous les objets du compartiment. Lorsque vous appliquez explicitement une période de rétention à une version d'objet, vous spécifiez une Retain Until Date (Date de fin de conservation) pour la version de l'objet. Amazon S3 stocke cette date dans les métadonnées de la version de l'objet.

Vous pouvez également définir une période de rétention dans les propriétés d'un compartiment. Lorsque vous définissez une période de rétention sur un compartiment, vous spécifiez une durée, en jours ou en années, pendant laquelle chaque version d'objet placée dans le compartiment sera protégée. Lorsque vous placez un objet dans le compartiment, Amazon S3 calcule une échéance de rétention pour la version de l'objet en ajoutant la durée spécifiée à l'horodatage de création de la version de l'objet. La version de l'objet est ensuite protégée exactement comme si vous aviez mis en place un verrouillage individuel de manière explicite avec la même période de rétention sur la version de l'objet.

> **Note**
>
> Lorsque vous placez (PUT) une version d'objet dotée d'un mode et d'une période de rétention individuels explicites dans un compartiment, les paramètres individuels de verrouillage d'objet de la version de l'objet remplacent tous les paramètres de rétention des propriétés du compartiment.

À l'instar de tous les autres paramètres de la fonctionnalité de verrouillage des objets, les périodes de rétention s'appliquent aux versions d'objet individuelles. Les différentes versions d'un seul objet peuvent avoir des modes et des périodes de rétention différents.

Par exemple, supposons que vous ayez un objet qui en est à 15 jours sur une période de conservation de 30 jours, et que vous PUT un objet dans Amazon S3 ayant le même nom et une période de rétention de 60 jours. Dans ce cas, votre demande PUT aboutit et Amazon S3 crée une nouvelle version de l'objet avec une période de rétention de 60 jours. L'ancienne version conservera sa période de rétention originale et pourra être supprimée dans 15 jours.

Après avoir appliqué un paramètre de rétention à une version d'objet, vous pouvez prolonger la période de rétention. Pour ce faire, envoyez une nouvelle demande de verrouillage d'objet pour la version d'objet avec une échéance de rétention postérieure à celle actuellement configurée pour la version d'objet. Amazon S3 remplace la période de rétention existante par la nouvelle période plus longue. Tout utilisateur disposant d'autorisations pour définir une période de rétention d'objet peut prolonger cette période pour une version d'objet. Pour définir une période de rétention, vous devez avoir l'autorisation `s3:PutObjectRetention`.

Lorsque vous définissez une période de rétention sur un objet ou un compartiment S3, vous devez sélectionner l'un des deux modes de rétention : conformité ou gouvernance.

Modes de conservation

Le verrouillage d'objet S3 fournit deux modes de rétention qui appliquent différents niveaux de protection à vos objets :

- Mode conformité
- Mode gouvernance

En mode Conformité, une version d'objet protégée ne peut pas être remplacée ou supprimée par n'importe quel utilisateur, notamment l'utilisateur racine de votre Compte AWS. Lorsqu'un objet est verrouillé en mode Conformité, son mode de rétention ne pas être modifié et sa période de rétention ne peut pas être raccourcie. Le mode Conformité garantit qu'une version d'objet ne peut pas être écrasée ou supprimée pendant la durée de la période de rétention.

> **Note**
>
> Le seul moyen de supprimer un objet en mode de conformité avant l'expiration de sa date de conservation est de supprimer l'objet associé Compte AWS.

Dans le mode Gouvernance, les utilisateurs ne peuvent pas remplacer ou supprimer une version d'objet ou en modifier les paramètres de verrouillage, sauf s'ils disposent d'autorisations spéciales. Avec le mode de gouvernance, vous protégez les objets contre leur suppression par la plupart des utilisateurs, mais vous pouvez toujours accorder à certains utilisateurs l'autorisation de modifier les paramètres de rétention ou de supprimer les objets si nécessaire. Vous pouvez également utiliser le mode Gouvernance pour tester les paramètres de la période de rétention, avant de créer une période de rétention en mode Conformité.

Pour remplacer ou supprimer des paramètres de rétention en mode de gouvernance, vous devez disposer de l'autorisation `s3:BypassGovernanceRetention` et explicitement inclure `x-amz-bypass-governance-retention:true` en tant qu'en-tête de toute demande nécessitant un remplacement dans le mode de gouvernance.

> **Note**
>
> Par défaut, la console Amazon S3 inclut l'en-tête `x-amz-bypass-governance-retention:true`. Si vous tentez de supprimer des objets protégés par le mode de gouvernance et que vous disposez de l'autorisation `s3:BypassGovernanceRetention`, l'opération aboutit.

Détentions légales

Avec le verrouillage d'objet, vous pouvez également définir une mise en suspens juridique sur une version d'objet. À l'instar d'une période de rétention, une détention légale empêche une version d'objet d'être remplacée ou supprimée. Toutefois, une mise en suspens juridique n'a pas de période de rétention associée et reste en vigueur jusqu'à sa suppression. Les détentions légales peuvent être librement mises en place et supprimées par tous les utilisateurs disposant de l'autorisation `s3:PutObjectLegalHold`.

Les détentions légales sont indépendantes des périodes de rétention. La mise en place d'une détention légale sur une version d'objet n'affecte pas le mode ou la période de rétention pour cette version d'objet.

Par exemple, supposons que vous placiez une mise en suspens juridique sur une version d'objet, alors que cette version d'objet est également protégée par une période de rétention. Si la période de rétention expire, l'objet ne perd pas sa protection WORM. À la place, la mise en suspens juridique continue de protéger l'objet jusqu'à ce qu'un utilisateur autorisé la supprime de manière explicite. De la même façon, si vous supprimez une détention légale alors qu'une période de rétention est en

vigueur pour une version d'objet, cette dernière reste protégée jusqu'à l'expiration de la période de rétention.

Bonnes pratiques d'utilisation de S3 Object Lock

Envisagez d'utiliser le mode Gouvernance si vous souhaitez empêcher la plupart des utilisateurs de supprimer des objets pendant une période de rétention prédéfinie, tout en laissant à certains utilisateurs disposant d'autorisations spéciales la possibilité de modifier les paramètres de rétention ou de supprimer les objets.

Envisagez d'utiliser le mode Conformité si vous ne souhaitez jamais qu'un utilisateur Compte AWS, y compris l'utilisateur root, puisse supprimer les objets pendant une période de rétention prédéfinie. Vous pouvez utiliser ce mode au cas où vous auriez besoin de stocker des données conformes.

Vous pouvez utiliser Legal Hold lorsque vous ne savez pas pendant combien de temps vous souhaitez que vos objets restent immuables. Cela peut être dû au fait que vous avez un audit externe de vos données à venir et que vous souhaitez que les objets restent immuables jusqu'à ce que l'audit soit terminé. Vous pouvez également avoir un projet en cours utilisant un ensemble de données que vous souhaitez conserver immuable jusqu'à ce que le projet soit terminé.

Autorisations nécessaires

Les opérations de verrouillage des objets nécessitent des autorisations spécifiques. En fonction de l'opération exacte que vous tentez de réaliser, vous aurez peut-être besoin de l'une des autorisations suivantes :

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`
- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Pour obtenir la liste complète des autorisations Amazon S3 avec leurs descriptions, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference.

Pour en savoir plus sur l'utilisation de conditions avec les autorisations, consultez Exemples de politiques relatives aux compartiments utilisant des clés de condition.

Considérations relatives au verrouillage d'objet

Le verrouillage d'objet Amazon S3 permet d'empêcher la suppression ou le remplacement d'objets sur une période déterminée ou indéfinie.

Vous pouvez utiliser la console Amazon S3, AWS Command Line Interface (AWS CLI), AWS les kits SDK ou l'API REST Amazon S3 pour afficher ou définir les informations de verrouillage des objets. Pour obtenir des informations générales sur les fonctionnalités de verrouillage d'objet S3, consultez Utilisation du verrouillage des objets S3.

> **Important**
>
> - Après avoir activé le verrouillage d'objet sur un compartiment, vous ne pouvez pas désactiver le verrouillage d'objet ni interrompre la gestion des versions pour ce compartiment.
> - Les compartiments S3 avec verrouillage d'objet ne peuvent pas être utilisés comme compartiments de destination pour les journaux d'accès au serveur. Pour plus d'informations, consultez the section called "Enregistrement de l'accès au serveur".

Rubriques

Autorisations d’affichage des informations de verrouillage

Vous pouvez afficher par programmation le statut de verrouillage d’objet d’une version d’objet Amazon S3 à l’aide des opérations HeadObject ou GetObject. Les deux opérations renvoient le mode de rétention, Rétention jusqu’à la date, et le statut de mise en suspens juridique pour la version d’objet spécifiée. En outre, vous pouvez consulter l'état du verrouillage des objets pour plusieurs objets de votre compartiment S3 à l'aide de S3 Inventory.

Pour afficher le mode de conservation et la période de conservation d'une version d'objet, vous devez avoir l'autorisation `s3:GetObjectRetention`. Pour afficher le statut de suspension juridique d'une version d'objet, vous devez avoir l'autorisation `s3:GetObjectLegalHold`. Pour afficher la configuration de rétention par défaut d’un compartiment, vous devez disposer de l’autorisation `s3:GetBucketObjectLockConfiguration`. Si vous effectuez une demande pour une configuration de verrouillage d’objet sur un compartiment pour lequel le verrouillage d’objet S3 n’est pas activé, Amazon S3 renvoie une erreur.

Ignorer le mode de gouvernance

Si vous disposez de l’autorisation `s3:BypassGovernanceRetention`, vous pouvez effectuer des opérations sur les versions d’objet verrouillées en mode de gouvernance comme si elles n’étaient pas protégées. Ces opérations incluent la suppression d’une version d’objet, le raccourcissement de la période de rétention ou la suppression de la période de rétention de verrouillage d’objet en plaçant une nouvelle demande `PutObjectRetention` avec des paramètres vides.

Afin d'ignorer le mode de gouvernance, vous devez indiquer explicitement dans votre demande que vous voulez l'ignorer. Pour ce faire, incluez l'`x-amz-bypass-governance-retention:true`en-tête dans votre demande d'opération d'`PutObjectRetention`API ou utilisez le paramètre équivalent pour les demandes effectuées via les AWS SDK AWS CLI ou. La console S3 applique automatiquement cet en-tête pour les demandes effectuées via la console S3 si vous disposez de l’autorisation `s3:BypassGovernanceRetention`.

> **Note**
>
> L'ignorance du mode de gouvernance n'affecte pas le statut de suspension juridique d'une version d'objet. Si une mise en suspens juridique est activée pour une version d’objet, cette mise en suspens reste en vigueur et empêche que les demandes remplacent ou suppriment la version d’objet.

Utilisation du verrouillage d’objet avec la réplication S3

Vous pouvez utiliser le verrouillage d’objet avec la réplication S3 pour activer la copie automatique et asynchrone d’objets verrouillés et de leurs métadonnées de rétention entre des compartiments S3. Cela signifie que pour les objets répliqués, Amazon S3 utilise la configuration de verrouillage des objets du compartiment source. En d'autres termes, si le verrouillage d'objet est activé dans le compartiment source, le verrouillage d'objet doit également être activé dans les compartiments de destination. Si un objet est directement chargé dans le compartiment de destination (en dehors de S3 Replication), il utilise le verrouillage d'objet défini sur le compartiment de destination. Lorsque vous utilisez la réplication, les objets d’un compartiment source sont répliqués vers un ou plusieurs compartiments de destination.

Pour configurer la réplication sur un compartiment dans lequel Object Lock est activé, vous pouvez utiliser la console S3 AWS CLI, l'API REST Amazon S3 ou AWS les kits SDK.

> **Note**
>
> Pour utiliser Object Lock avec la réplication, vous devez accorder deux autorisations supplémentaires sur le compartiment S3 source dans le rôle AWS Identity and Access Management (IAM) que vous utilisez pour configurer la réplication. Les deux autorisations supplémentaires sont `s3:GetObjectRetention` et `s3:GetObjectLegalHold`. Si le rôle dispose d’une déclaration d’autorisation `s3:Get*`, cette déclaration répond à l’exigence. Pour plus d’informations, consultez Configuration des autorisations pour la réplication en direct.
>
> Pour obtenir des informations générales sur la réplication S3, consultez Vue d'ensemble de la réplication d'objets.
>
> Pour examiner des exemples de configuration de la réplication S3, consultez Exemples de configuration de la réplication en direct.

Utilisation du verrouillage d’objet avec l’inventaire Amazon S3

Vous pouvez configurer l’inventaire Amazon S3 pour créer des listes d’objets dans un compartiment S3 selon une planification définie. Vous pouvez configurer l’inventaire Amazon S3 pour inclure les métadonnées de verrouillage d’objet suivantes pour vos objets :

- Date limite de rétention
- Mode de rétention

- Statut de mise en suspens juridique

Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).

Gestion des politiques de cycle de vie S3 avec Object Lock

Les configurations de gestion du cycle de vie des objets continuent à fonctionner normalement sur les objets protégés, y compris le placement des marqueurs de suppression. Cependant, une version verrouillée d'un objet ne peut pas être supprimée par une politique d'expiration du cycle de vie S3. Object Lock est maintenu quelle que soit la classe de stockage dans laquelle réside l'objet et tout au long des transitions entre les classes de stockage du cycle de vie S3.

Pour plus d'informations sur la gestion des cycles de vie d'objet, consultez Gestion du cycle de vie de votre stockage.

Gestion des marqueurs de suppression avec Object Lock

Même si vous ne pouvez pas supprimer une version d'objet protégée, vous pouvez continuer à créer un marqueur de suppression pour cet objet. Le placement d'un marqueur de suppression sur un objet n'a pas pour effet de supprimer l'objet ou des versions de celui-ci. Cependant, il conduit Amazon S3 à se comporter la plupart du temps comme si l'objet avait été supprimé. Pour plus d'informations, consultez Utilisation des marqueurs de suppression.

> **Note**
>
> Les marqueurs de suppression ne sont pas protégés par WORM, quelles que soient la période de conservation ou la suspension juridique en place sur l'objet sous-jacent.

Utilisation de S3 Storage Lens avec le verrouillage d'objet

Pour consulter les statistiques relatives aux nombres d'objets et d'octets de stockage avec verrouillage d'objets activé, vous pouvez utiliser Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets.

Pour plus d'informations, consultez Utiliser S3 Storage Lens pour protéger vos données.

Pour obtenir une liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.

Téléchargement d'objets vers un bucket activé par Object Lock

L'`Content-MD5`en-tête est obligatoire pour toute demande de téléchargement d'un objet dont la période de rétention est configurée à l'aide d'Object Lock. Le condensé MD5 permet de vérifier l'intégrité de votre objet après l'avoir chargé dans un bucket. Après avoir chargé l'objet, Amazon S3 calcule le récapitulatif MD5 de l'objet et le compare à la valeur que vous avez fournie. La requête n'aboutit que si les deux récapitulatifs correspondent. La console S3 ajoute automatiquement cet en-tête, mais vous devez le spécifier lorsque vous utilisez l'PutObjectAPI.

Pour plus d'informations, consultez Utilisation de Content-MD5 pour charger des objets.

Configuration d'événements et de notifications

Vous pouvez utiliser les notifications d'événements Amazon S3 pour suivre l'accès et les modifications apportées à vos configurations et données Object Lock en utilisant AWS CloudTrail. Pour plus d'informations CloudTrail, voir Qu'est-ce que c'est AWS CloudTrail ? dans le guide de AWS CloudTrail l'utilisateur.

Vous pouvez également utiliser Amazon CloudWatch pour générer des alertes sur la base de ces données. Pour plus d'informations CloudWatch, consultez le document Qu'est-ce qu'Amazon CloudWatch ? dans le guide de CloudWatch l'utilisateur Amazon.

Définition de limites sur les périodes de rétention à l'aide d'une politique de compartiment

Vous pouvez définir des périodes de rétention autorisées minimale et maximale pour un compartiment en utilisant une politique de compartiment. La période de conservation maximale est de 100 ans.

L'exemple suivant montre une stratégie de compartiment qui utilise la clé de condition `s3:object-lock-remaining-retention-days` pour définir une période de conservation maximale de 10 jours.

```
{
    "Version": "2012-10-17",
    "Id": "SetRetentionLimits",
    "Statement": [
        {
            "Sid": "SetRetentionPeriod",
            "Effect": "Deny",
            "Principal": "*",
```

```
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": "arn:aws:s3:::example-s3-bucket1/*",
            "Condition": {
                "NumericGreaterThan": {
                    "s3:object-lock-remaining-retention-days": "10"
                }
            }
        }
    ]
}
```

> **Note**
>
> Si votre compartiment est le compartiment de destination d'une configuration de réplication, vous pouvez configurer des périodes de rétention minimale et maximale autorisées pour les réplicas d'objet créés à l'aide de la réplication. Pour ce faire, vous devez autoriser l'action `s3:ReplicateObject` dans votre politique de compartiment. Pour plus d'informations sur les autorisations de réplication, consultez the section called "Configurer les autorisations".

Pour plus d'informations sur les politiques de compartiment, consultez les rubriques suivantes :

- Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference
- Opérations sur les objets
- Exemples de politiques relatives aux compartiments utilisant des clés de condition

Configuration du verrouillage d'objet S3

Avec Amazon S3 Object Lock, vous pouvez stocker des objets dans Amazon S3 à l'aide d'un modèle write-once-read-many(WORM). En utilisant le verrouillage des objets S3, vous pouvez empêcher qu'un objet ne soit supprimé ou remplacé sur une période déterminée ou indéfinie. Pour obtenir des informations générales sur les fonctionnalités du verrouillage d'objet, consultez Utilisation du verrouillage des objets S3.

Avant de verrouiller des objets, vous devez activer la gestion des versions S3 et le verrouillage d'objet sur un compartiment. Vous pouvez ensuite définir une période de rétention, une mise en suspens juridique ou les deux.

Pour utiliser le verrouillage d’objet, vous devez disposer de certaines autorisations. Pour obtenir la liste des autorisations associées aux différentes opérations de verrouillage d’objet, consultez the section called “Autorisations nécessaires”.

> **Important**
>
> - Après avoir activé le verrouillage d’objet sur un compartiment, vous ne pouvez pas désactiver le verrouillage d’objet ni interrompre la gestion des versions pour ce compartiment.
> - Les compartiments S3 avec verrouillage d’objet ne peuvent pas être utilisés comme compartiments de destination pour les journaux d’accès au serveur. Pour plus d’informations, consultez the section called “Enregistrement de l'accès au serveur”.

Rubriques

Activation du verrouillage d’objet à la création d’un nouveau compartiment S3

Vous pouvez activer Object Lock lors de la création d'un nouveau compartiment S3 à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI), AWS des SDK ou de l'API REST Amazon S3.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez Créer un compartiment.

   La page Créer un compartiment s'ouvre.
4. Pour Nom du compartiment, saisissez le nom de votre compartiment.

> **Note**
>
> Après avoir créé un compartiment, vous ne pouvez pas modifier son nom. Pour plus d'informations sur l'attribution de noms à des compartiments, consultez Règles de dénomination de compartiment.

5. Pour Région, choisissez l' Région AWS endroit où vous souhaitez que le compartiment réside.
6. Sous Propriété d'objets, choisissez de désactiver ou d'activer les listes de contrôle d'accès (ACL) et de contrôler la propriété des objets chargés dans votre compartiment.
7. Dans Paramètres de blocage de l'accès public pour ce compartiment, choisissez les paramètres Bloquer l'accès public que vous souhaitez appliquer au compartiment.
8. Sous Gestion des versions de compartiment, choisissez Activée.

   Le verrouillage d'objet fonctionne uniquement avec des compartiments versionnés.
9. (Facultatif) Sous Tags (Balises), vous pouvez choisir d'ajouter des balises à votre compartiment. Les balises sont des paires clé-valeur utilisées pour classer le stockage et allouer des coûts.
10. Sous Paramètres avancés, recherchez Verrouillage d'objet et choisissez Activer.

    Vous devez reconnaître que l'activation du verrouillage d'objet permettra de verrouiller définitivement les objets de ce compartiment.
11. Choisissez Créer un compartiment.

À l'aide du AWS CLI

L'exemple `create-bucket` suivant crée un nouveau compartiment S3 nommé *`example-s3-bucket1`* avec le verrouillage d'objet activé :

```
aws s3api create-bucket --bucket example-s3-bucket1 --object-lock-enabled-for-bucket
```

Pour plus d'informations et des exemples, consultez create-bucket dans la Référence des commandes AWS CLI .

> **Note**
>
> Vous pouvez exécuter AWS CLI des commandes depuis la console en utilisant AWS CloudShell. AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous

pouvez lancer directement depuis le. AWS Management Console Pour plus d'informations, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour créer un nouveau compartiment S3 avec le verrouillage d'objet activé. Pour plus d'informations, veuillez consulter CreateBucket dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS kits de développement logiciel

Pour des exemples expliquant comment activer Object Lock lors de la création d'un nouveau compartiment S3 avec AWS les SDK, consultezUtilisation `CreateBucket` avec un AWS SDK ou une CLI.

Pour des exemples expliquant comment obtenir la configuration actuelle de Object Lock avec les AWS SDK, consultezUtilisation `GetObjectLockConfiguration` avec un AWS SDK ou une CLI.

Pour un scénario interactif illustrant les différentes fonctionnalités d'Object Lock à l'aide AWS des SDK, voirUtilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK.

Pour obtenir des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Activation du verrouillage d'objet sur un compartiment S3 existant

Vous pouvez activer Object Lock pour un compartiment S3 existant à l'aide de la console Amazon S3 AWS CLI, AWS des SDK ou de l'API REST Amazon S3.

Utilisation de la console S3

> **Note**
>
> Le verrouillage d'objet fonctionne uniquement avec des compartiments versionnés.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.

3. Dans la liste Compartiments, choisissez le nom du compartiment sur lequel vous souhaitez activer le verrouillage d'objet.
4. Choisissez l'onglet Propriétés.
5. Sous Propriétés, faites défiler l'affichage jusqu'à la section Verrouillage d'objet et choisissez Modifier.
6. Sous Verrouillage d'objet, choisissez Activer.

   Vous devez reconnaître que l'activation du verrouillage d'objet permettra de verrouiller définitivement les objets de ce compartiment.
7. Sélectionnez Enregistrer les modifications.

À l'aide du AWS CLI

L'exemple de commande `put-object-lock-configuration` suivant définit une période de rétention de verrouillage d'objet de 50 jours sur un compartiment nommé *`example-s3-bucket1`* :

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Pour plus d'informations et des exemples, consultez put-object-lock-configuration dans la Référence des commandes AWS CLI .

> **Note**
>
> Vous pouvez exécuter AWS CLI des commandes depuis la console en utilisant AWS CloudShell. AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console Pour plus d'informations, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST Amazon S3 pour activer le verrouillage d'objet sur un compartiment S3 existant. Pour plus d'informations, veuillez consulter PutObjectLockConfiguration dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS kits de développement logiciel

Pour des exemples d'activation du verrouillage d'objets pour un compartiment S3 existant à l'aide AWS des SDK, consultezUtilisation `PutObjectLockConfiguration` avec un AWS SDK ou une CLI.

Pour des exemples expliquant comment obtenir la configuration actuelle de Object Lock avec les AWS SDK, consultezUtilisation `GetObjectLockConfiguration` avec un AWS SDK ou une CLI.

Pour un scénario interactif illustrant les différentes fonctionnalités d'Object Lock à l'aide AWS des SDK, voirUtilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK.

Pour obtenir des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Définition ou modification d’une mise en suspens juridique sur un objet S3

Vous pouvez définir ou supprimer un blocage légal sur un objet S3 à l'aide de la console Amazon S3 AWS CLI, des AWS SDK ou de l'API REST Amazon S3.

> **Important**
>
> - Si vous souhaitez définir une mise en suspens juridique sur un objet, le verrouillage d’objet doit déjà être activé sur le compartiment de l’objet.
> - Lorsque vous placez (PUT) une version d’objet dotée d’un mode et d’une période de rétention individuels explicites dans un compartiment, les paramètres individuels de verrouillage d’objet de la version de l’objet remplacent tous les paramètres de rétention des propriétés du compartiment.

Pour plus d’informations, consultez the section called “Détentions légales”.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment qui contient l’objet sur lequel vous souhaitez définir ou modifier une mise en suspens juridique.

4. Dans la liste Objets, sélectionnez l'objet sur lequel vous souhaitez définir ou modifier une mise en suspens juridique.
5. Sur la page Propriétés de l'objet, recherchez la section Conservation légale du verrouillage d'objet et choisissez Modifier.
6. Choisissez Activer pour définir une mise en suspens juridique ou Désactiver pour supprimer une mise en suspens juridique.
7. Sélectionnez Enregistrer les modifications.

À l'aide du AWS CLI

L'exemple `put-object-legal-hold` suivant définit une mise en suspens juridique sur l'objet *`my-image.fs`* dans le compartiment nommé *`example-s3-bucket1`* :

```
aws s3api put-object-legal-hold --bucket example-s3-bucket1 --key my-image.fs --legal-hold="Status=ON"
```

L'exemple `put-object-legal-hold` suivant supprime une mise en suspens juridique sur l'objet *`my-image.fs`* dans le compartiment nommé *`example-s3-bucket1`* :

```
aws s3api put-object-legal-hold --bucket example-s3-bucket1 --key my-image.fs --legal-hold="Status=OFF"
```

Pour plus d'informations et des exemples, consultez put-object-legal-hold dans la Référence des commandes AWS CLI .

> **Note**
>
> Vous pouvez exécuter AWS CLI des commandes depuis la console en utilisant AWS CloudShell. AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console Pour plus d'informations, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour définir ou modifier une mise en suspens juridique sur un objet. Pour plus d'informations, veuillez consulter PutObjectLegalHold dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS kits de développement logiciel

Pour des exemples sur la manière de définir une conservation légale sur un objet à l'aide AWS des SDK, consultezUtilisation `PutObjectLegalHold` avec un AWS SDK ou une CLI.

Pour des exemples expliquant comment obtenir le statut de blocage légal actuel avec les AWS SDK, consultezObtenez la configuration de conservation légale d'un objet Amazon S3 à l'aide d'un AWS SDK.

Pour un scénario interactif illustrant les différentes fonctionnalités d'Object Lock à l'aide AWS des SDK, voirUtilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK.

Pour obtenir des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Définition ou modification d’une période de rétention sur un objet S3

Vous pouvez définir ou modifier une période de rétention sur un objet S3 à l'aide de la console Amazon S3 AWS CLI, des AWS kits SDK ou de l'API REST Amazon S3.

> **Important**
>
> - Si vous souhaitez définir une période de rétention sur un objet, le verrouillage d’objet doit déjà être activé sur le compartiment de l’objet.
> - Lorsque vous placez (PUT) une version d’objet dotée d’un mode et d’une période de rétention individuels explicites dans un compartiment, les paramètres individuels de verrouillage d’objet de la version de l’objet remplacent tous les paramètres de rétention des propriétés du compartiment.
> - Le seul moyen de supprimer un objet en mode de conformité avant l'expiration de sa date de conservation est de supprimer l'objet associé Compte AWS.

Pour plus d’informations, consultez Périodes de rétention.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment qui contient l'objet sur lequel vous souhaitez définir ou modifier une période de rétention.
4. Dans la liste Objets, sélectionnez l'objet sur lequel vous souhaitez définir ou modifier une période de rétention.
5. Sur la page Propriétés de l'objet, recherchez la section Rétention Verrouillage d'objet et choisissez Modifier.
6. Sous Rétention, choisissez Activer pour définir une période de rétention ou Désactiver pour supprimer une période de rétention.
7. Si vous avez choisi Activer, sous Mode de rétention, choisissez Mode de gouvernance ou Mode de conformité. Pour plus d'informations, consultez Modes de conservation.
8. Sous Rétention jusqu'à la date, choisissez la date à laquelle vous souhaitez que la période de rétention se termine. Durant cette période, l'objet est protégé selon le modèle WORM et ne peut être ni remplacé ni supprimé. Pour plus d'informations, consultez Périodes de rétention.
9. Choisissez Enregistrer les modifications.

À l'aide du AWS CLI

L'exemple `put-object-retention` suivant définit une période de rétention sur l'objet *my-image.fs* dans le compartiment nommé *example-s3-bucket1* jusqu'au 1er janvier 2025 :

```
aws s3api put-object-retention --bucket example-s3-bucket1 --key my-image.fs --
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Pour plus d'informations et des exemples, consultez put-object-retention dans la Référence des commandes AWS CLI .

> **Note**
>
> Vous pouvez exécuter AWS CLI des commandes depuis la console en utilisant AWS CloudShell. AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console Pour plus d'informations, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour définir une période de rétention sur un objet. Pour plus d'informations, veuillez consulter PutObjectRetention dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS kits de développement logiciel

Pour des exemples de définition d'une période de rétention pour un objet à l'aide AWS des SDK, consultezUtilisation `PutObjectRetention` avec un AWS SDK ou une CLI.

Pour des exemples expliquant comment obtenir la période de rétention d'un objet à l'aide AWS des SDK, consultezUtilisation `GetObjectRetention` avec un AWS SDK ou une CLI.

Pour un scénario interactif illustrant les différentes fonctionnalités d'Object Lock à l'aide AWS des SDK, voirUtilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK.

Pour obtenir des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Définition ou modification d'une période de rétention par défaut sur un compartiment S3

Vous pouvez définir ou modifier une période de rétention par défaut sur un compartiment S3 à l'aide de la console Amazon S3 AWS CLI, des AWS kits SDK ou de l'API REST Amazon S3. Vous spécifiez une durée, en jours ou en années, pendant laquelle chaque version d'objet placée dans le compartiment sera protégée.

Important

- Si vous souhaitez définir une période de rétention par défaut sur un compartiment, le verrouillage d'objet doit déjà être activé sur ce compartiment.
- Lorsque vous placez (PUT) une version d'objet dotée d'un mode et d'une période de rétention individuels explicites dans un compartiment, les paramètres individuels de verrouillage d'objet de la version de l'objet remplacent tous les paramètres de rétention des propriétés du compartiment.
- Le seul moyen de supprimer un objet en mode de conformité avant l'expiration de sa date de conservation est de supprimer l'objet associé Compte AWS.

Pour plus d'informations, consultez Périodes de rétention.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment sur lequel vous souhaitez définir ou modifier une période de rétention par défaut.
4. Choisissez l'onglet Propriétés.
5. Sous Propriétés, faites défiler l'affichage jusqu'à la section Verrouillage d'objet et choisissez Modifier.
6. Sous Rétention par défaut, choisissez Activer pour définir une rétention par défaut ou Désactiver pour supprimer une rétention par défaut.
7. Si vous avez choisi Activer, sous Mode de rétention, choisissez Mode de gouvernance ou Mode de conformité. Pour plus d'informations, consultez Modes de conservation.
8. Sous Période de rétention par défaut, choisissez le nombre de jours ou d'années que vous souhaitez comme période de rétention. Les objets placés dans ce compartiment seront verrouillés pendant ce nombre de jours ou d'années. Pour plus d'informations, consultez Périodes de rétention.
9. Choisissez Enregistrer les modifications.

À l'aide du AWS CLI

L'exemple de commande `put-object-lock-configuration` suivant définit une période de rétention de verrouillage d'objet de 50 jours sur le compartiment nommé *`example-s3-bucket1`* en utilisant le mode de conformité :

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

L'exemple `put-object-lock-configuration` suivant supprime la configuration de rétention par défaut sur un compartiment :

```
aws s3api put-object-lock-configuration --bucket example-s3-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled"}'
```

Pour plus d'informations et des exemples, consultez put-object-lock-configuration dans la Référence des commandes AWS CLI .

> **Note**
>
> Vous pouvez exécuter AWS CLI des commandes depuis la console en utilisant AWS CloudShell. AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console Pour plus d'informations, voir Qu'est-ce que c'est CloudShell ? dans le guide de AWS CloudShell l'utilisateur.

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour définir une période de rétention par défaut sur un compartiment S3 existant. Pour plus d'informations, veuillez consulter PutObjectLockConfiguration dans la Référence d'API Amazon Simple Storage Service.

Utilisation des AWS kits de développement logiciel

Pour des exemples de définition d'une période de rétention par défaut sur un compartiment S3 existant à l'aide AWS des SDK, consultezUtilisation `PutObjectLockConfiguration` avec un AWS SDK ou une CLI.

Pour un scénario interactif illustrant les différentes fonctionnalités d'Object Lock à l'aide AWS des SDK, voirUtilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK.

Pour obtenir des informations générales sur l'utilisation des différents AWS SDK, consultezDéveloppement avec Amazon S3 à l'aide des AWS SDK.

Utilisation des classes de stockage Simple Storage Service (Amazon S3)

Dans Simple Storage Service (Amazon S3), chaque objet possède une classe de stockage qui lui est associée. Par exemple, si vous listez les objets dans un compartiment S3, la console montre la classe de stockage pour tous les objets dans la liste. Simple Storage Service (Amazon S3) offre une plage de classes de stockage pour les objets que vous stockez. Vous choisissez une classe

de stockage en fonction du cas d'utilisation et des exigences en matière de performances d'accès. Toutes ces classes de stockage offrent une durabilité élevée.

Les sections suivantes fournissent des détails sur les différentes classes de stockage et vous expliquent comment définir une classe de stockage pour vos objets.

Rubriques

Classes de stockage pour les objets fréquemment consultés

Pour les cas d'utilisation sensibles aux performances (requérant des temps d'accès de l'ordre de la milliseconde) et les données fréquemment consultées, Simple Storage Service (Amazon S3) fournit les classes de stockage suivantes :

- S3 Standard : classe de stockage par défaut. Si vous ne spécifiez pas la classe de stockage lors du chargement d'un objet, Simple Storage Service (Amazon S3) affecte la classe de stockage S3 Standard.
- S3 Express One Zone : Amazon S3 Express One Zone est une classe de stockage Amazon S3 d’une seule zone à hautes performances, spécialement conçue pour fournir un accès aux données cohérent en moins de dix millisecondes pour vos applications les plus sensibles à la latence. S3 Express One Zone est la classe de stockage d'objets cloud à latence la plus faible disponible à ce jour, avec une vitesse d'accès aux données jusqu'à 10 fois plus rapide et des coûts de demande 50 % inférieurs à ceux de S3 Standard. Avec S3 Express One Zone, vos données sont stockées de façon redondante sur plusieurs appareils au sein d’une même zone de disponibilité. Pour plus d’informations, consultez Qu’est-ce que S3 Express One Zone ?.
- Redondance réduite : la classe de stockage à redondance réduite (RRS) est conçue pour les données reproductibles non critiques pouvant être stockées avec moins de redondance que la classe de stockage S3 standard.

> **Important**
>
> Nous vous recommandons de ne pas utiliser cette classe de stockage. La classe de stockage S3 Standard est plus économique.

En matière de durabilité, les objets RRS présentent une perte moyenne annuelle de 0,01 % d'objets. Si un objet RRS est perdu, lorsque des demandes sont effectuées sur cet objet, Simple Storage Service (Amazon S3) renvoie une erreur 405.

Classe de stockage pour l'optimisation automatique des données avec des modèles d'accès inconnus ou irréguliers

S3 Intelligent-Tiering est une classe de stockage Amazon S3 conçue pour optimiser les coûts de stockage en déplaçant automatiquement les données vers le niveau d'accès le plus rentable, sans impact sur les performances ni sur les coûts opérationnels. S3 Intelligent-Tiering est la seule classe de stockage dans le cloud qui permet de réaliser automatiquement des économies en déplaçant les données à un niveau d'objet granulaire entre les niveaux d'accès lorsque les modèles d'accès changent. S3 Intelligent-Tiering est la classe de stockage idéale si vous souhaitez optimiser vos coûts de stockage pour les données dont les modèles d'accès sont inconnus ou variables. L'utilisation de la classe S3 Intelligent-Tiering n'implique aucuns frais de récupération.

Moyennant des frais mensuels minimes de surveillance et d'automatisation des objets, S3 Intelligent-Tiering surveille les schémas d'accès et déplace automatiquement les objets qui n'ont pas été consultés vers des niveaux d'accès moins coûteux. S3 Intelligent-Tiering permet de réduire automatiquement les coûts de stockage grâce à trois niveaux d’accès à faible latence et à haut débit. Pour les données accessibles de manière asynchrone, vous pouvez choisir d'activer les capacités d'archivage automatique dans la classe de stockage S3 Intelligent-Tiering. S3 Intelligent-Tiering est conçue pour une disponibilité de 99,9 % et une durabilité de 99,9999999 %.

S3 Intelligent-Tiering stocke automatiquement les objets dans trois niveaux d'accès :

- Accès fréquent : les objets chargés ou migrés dans S3 Intelligent-Tiering sont stockés automatiquement au niveau Accès fréquent.
- Accès peu fréquent : S3 Intelligent-Tiering déplace vers le niveau Accès peu fréquent les objets qui n'ont pas été consultés pendant 30 jours consécutifs.

- Accès Archive Instant : avec S3 Intelligent-Tiering, tous les objets existants qui n'ont pas été consultés pendant 90 jours consécutifs sont automatiquement transférés vers le niveau d'accès Archive Instant.

Outre ces trois niveaux, S3 Intelligent-Tiering propose deux niveaux d'accès d'archive en option :

- Accès Archive : S3 Intelligent-Tiering vous offre la possibilité d'activer le niveau d'accès Archive pour les données accessibles de manière asynchrone. Une fois activé, le niveau d'accès Archive archive automatiquement les objets qui n'ont pas été consultés pendant un minimum de 90 jours consécutifs.
- Accès Deep Archive : S3 Intelligent-Tiering vous offre la possibilité d'activer le niveau d'accès Deep Archive pour les données accessibles de manière asynchrone. Une fois activé, le niveau d'accès Deep Archive archive automatiquement les objets qui n'ont pas été consultés pendant un minimum de 180 jours consécutifs.

> **Note**
>
> - N'activez le niveau d'accès Archive que pendant 90 jours si vous souhaitez contourner le niveau d'accès Archive Instant. Le niveau Archive Access permet un stockage légèrement moins coûteux avec des délais de minute-to-hour récupération. Le niveau d'accès Archive Instant offre un accès de l'ordre de la milliseconde et des performances à haut débit.
> - Activez les niveaux d'accès Archive et Deep Archive uniquement si vos objets sont accessibles de manière asynchrone par votre application. Si l'objet que vous récupérez est stocké au niveau d'accès Archive ou Deep Archive, restaurez d'abord l'objet à l'aide de `RestoreObject`.

Vous pouvez déplacer les données récemment créées vers S3 Intelligent-Tiering en le définissant comme classe de stockage par défaut. Vous pouvez également choisir d'activer l'un ou les deux niveaux d'accès aux archives à l'aide de l'opération `PutBucketIntelligentTieringConfiguration`API, de la AWS CLI console Amazon S3 ou des deux. Pour plus d'informations sur l'utilisation de S3 Intelligent-Tiering et sur l'activation des niveaux d'accès d'archive, consultez Utiliser S3 Intelligent-Tiering.

Pour accéder à des objets aux niveaux Archive Access ou Deep Archive Access, vous devez d'abord les restaurer. Pour plus d'informations, consultez Restauration des objets à partir des niveaux d'accès Archive et Deep Archive de S3 Intelligent-Tiering.

> **Note**
>
> Si la taille d'un objet est inférieure à 128 Ko, il ne sera pas admissible à la hiérarchisation automatique. Les objets plus petits sont toujours stockés dans le niveau Accès fréquent. Pour plus d'informations sur S3 Intelligent-Tiering, consultez Niveaux d'accès S3 Intelligent-Tiering..

Classes de stockage pour les objets à accès peu fréquent

Les classes de stockage S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent sont conçues pour des données à longue durée de vie et à accès peu fréquent. (IA correspond à « Infrequent Access » [Accès peu fréquent].) Les objets S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent sont disponibles avec un temps d'accès de l'ordre de la milliseconde (comme la classe de stockage S3 Standard). Simple Storage Service (Amazon S3) facture des frais de récupération pour ces objets. Ils sont donc plus adaptés aux données à accès peu fréquent. Pour en savoir plus sur la tarification, veuillez consulter Tarification Simple Storage Service (Amazon S3).

Par exemple, vous pouvez choisir les classes de stockage S3 Standard – Accès peu fréquent et S3 unizone – Accès peu fréquent pour les utilisations suivantes :

- Pour le stockage des sauvegardes.
- Pour des données anciennes qui sont rarement consultées, mais qui requièrent encore des temps d'accès de l'ordre de la milliseconde. Par exemple, lorsque vous chargez des données, vous pouvez choisir la classe de stockage S3 standard et utiliser la configuration du cycle de vie pour indiquer à Simple Storage Service (Amazon S3) de faire passer les objets vers la classe S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.

  Pour plus d'informations sur la gestion du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Les classes de stockage S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent sont adaptées pour les objets de plus de 128 Ko que vous envisagez de stocker pendant au moins 30 jours. Si un objet a une taille inférieure à 128 Ko, Amazon S3 vous facture 128 Ko. Si vous supprimez un objet avant la période minimale de stockage de 30 jours, les 30 jours vous sont facturés. Les objets supprimés, remplacés ou transférés dans une autre classe de stockage dans les 30 jours sont soumis aux frais d'utilisation normaux du stockage, auxquels s'ajoutent des frais calculés au prorata pour le reste de la période minimale de 30 jours. Pour en savoir plus sur la tarification, veuillez consulter Tarification Amazon S3.

Ces classes de stockage diffèrent sur les points suivants :

- S3 standard – Accès peu fréquent : Amazon S3 stocke les données d'objet de manière redondante dans plusieurs zones de disponibilité géographiquement séparées les unes des autres (comme la classe de stockage S3 Standard). Les objets S3 standard – Accès peu fréquent résistent à la perte d'une zone de disponibilité. Cette classe de stockage offre plus de disponibilité et de résilience que la classe S3 unizone – Accès peu fréquent.
- S3 unizone – Accès peu fréquent : Amazon S3 stocke les données d'objet dans une seule zone de disponibilité, ce qui en fait une solution moins onéreuse que S3 Standard – Accès peu fréquent. Toutefois, les données ne résistent pas à la perte physique de la zone de disponibilité suite à une catastrophe naturelle, telle qu'un séisme ou une inondation. La classe de stockage S3 unizone – Accès peu fréquent a la même durabilité que S3 Standard – Accès peu fréquent, mais elle présente une disponibilité et une résilience moindres. Pour comparer la durabilité et la disponibilité des classes de stockage, consultez Comparaison des classes de stockage Amazon S3 à la fin de cette section. Pour en savoir plus sur la tarification, veuillez consulter Tarification Amazon S3.

Nous vous recommandons la procédure suivante :

- S3 standard – Accès peu fréquent : utilisez cette classe pour votre copie principale ou unique de données qui ne peuvent pas être recréées.
- S3 unizone – Accès peu fréquent : utilisez cette classe si vous pouvez recréer les données en cas de défaillance de la zone de disponibilité et pour les réplicas d'objets lorsque vous configurez la réplication entre régions (CRR) S3.

Classes de stockage pour les objets rarement consultés

Les classes de stockage S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont conçues pour le stockage et l'archivage des données à long terme à faible coût. Ces classes de stockage offrent une durabilité et une résilience égales à celles des classes de stockage S3 standard et S3 Standard – Accès peu fréquent. Pour plus d'informations sur les classes de stockage S3 Glacier, consultezStockage de données à long terme avec les classes de stockage S3 Glacier.

Amazon S3 fournit les classes de stockage S3 Glacier suivantes :

- S3 Glacier Instant Retrieval : à utiliser pour les données à long terme rarement consultées et nécessitant une extraction en quelques millisecondes. Les données de cette classe de stockage sont accessibles en temps réel.
- S3 Glacier Flexible Retrieval : utilisez cette classe pour les archives où des portions de données doivent être récupérées en quelques minutes. Les données de cette classe de stockage sont archivées et ne sont pas accessibles en temps réel.
- S3 Glacier Deep Archive : utilisez cette classe pour archiver les données qui ont rarement besoin d'être consultées. Les données de cette classe de stockage sont archivées et ne sont pas accessibles en temps réel.

Récupération d'objets archivés

Vous pouvez définir la classe de stockage d'un objet sur S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive de la même façon que vous le feriez pour les autres classes de stockage, tel que décrit dans la section Définition de la classe de stockage d'un objet. Toutefois, les objets S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont archivés et ne sont pas accessibles en temps réel. Pour plus d’informations, consultez Stockage d'archives.

> **Note**
>
> Lorsque vous utilisez les classes de stockage S3 Glacier, vos objets restent dans Amazon S3. Vous ne pouvez pas y accéder directement via le service Amazon S3 Glacier distinct. Pour plus d'informations sur le service Amazon S3 Glacier, consultez le guide du développeur Amazon S3 Glacier.

Classe de stockage pour Amazon S3 sur Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos AWS Outposts ressources et stocker et récupérer des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. Vous pouvez utiliser les mêmes opérations et fonctionnalités d'API AWS Outposts que sur Amazon S3, notamment les politiques d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts via les AWS Management Console AWS SDK ou AWS CLI l'API REST.

S3 sur Outposts fournit une nouvelle classe de stockage, S3 Outposts (`OUTPOSTS`). La classe de stockage S3 Outposts n'est disponible que pour les objets stockés dans des compartiments sur Outposts. Si vous essayez d'utiliser cette classe de stockage avec un compartiment S3 dans un Région AWS, une `InvalidStorageClass` erreur se produit. En outre, si vous essayez d'utiliser d'autres classes de stockage S3 avec des objets stockés dans S3 on Outposts, la même réponse d'erreur se produit.

Les objets stockés dans la classe de stockage S3 Outposts (`OUTPOSTS`) sont toujours chiffrés à l'aide du chiffrement côté serveur avec les clés de chiffrement gérées par Amazon S3 (SSE-S3). Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

Vous pouvez également choisir explicitement de chiffrer les objets stockés dans la classe de stockage S3 Outposts en utilisant le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C). Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C).

> **Note**
>
> S3 on Outposts ne prend pas en charge le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS).

Pour plus d'informations sur S3 on Outposts, consultez Qu'est-ce que Amazon S3 sur Outposts ?.

Comparaison des classes de stockage Amazon S3

Le tableau suivant compare les classes de stockage, y compris leur disponibilité, leur durabilité, leur durée de stockage minimale et d'autres considérations.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

* S3 Glacier Flexible Retrieval nécessite 40 Ko de métadonnées supplémentaires pour chaque objet archivé. Cela inclut 32 Ko de métadonnées facturés au tarif S3 Glacier Flexible Retrieval (requis pour identifier et récupérer vos données), et 8 Ko de données supplémentaires facturés au tarif S3 Standard. Le tarif S3 Standard est requis pour conserver le nom et les métadonnées définis par l'utilisateur pour les objets archivés dans S3 Glacier Flexible Retrieval. Pour plus d'informations sur les classes de stockage, consultez Classes de stockage Amazon S3.

** S3 Glacier Deep Archive nécessite 40 Ko de métadonnées supplémentaires pour chaque objet archivé. Cela inclut 32 Ko de métadonnées facturés au tarif S3 Glacier Deep Archive (requis pour identifier et récupérer vos données), et 8 Ko de données supplémentaires facturés au tarif S3 Standard. Le tarif S3 Standard est requis pour conserver le nom et les métadonnées définis par l'utilisateur pour les objets archivés dans Amazon S3 Glacier Deep Archive. Pour plus d'informations sur les classes de stockage, consultez Classes de stockage Amazon S3.

Sachez que toutes les classes de stockage à l’exception de S3 One Zone-IA et S3 Express One Zone sont conçues pour résister à la perte physique d’une zone de disponibilité suite à une catastrophe naturelle. De plus, outre les performances requises par votre scénario d'application, tenez compte des coûts. Pour connaître les prix des classes de stockage, veuillez consulter Tarification Amazon S3.

Définition de la classe de stockage d'un objet

Pour définir et mettre à jour les classes de stockage d'objets, vous pouvez utiliser la console Amazon S3, AWS les SDK ou le AWS Command Line Interface (AWS CLI). Toutes ces approches utilisent les opérations d'API Amazon S3 pour envoyer des demandes à Amazon S3.

Les opérations d'API Amazon S3 prennent en charge le paramétrage (ou la mise à jour) de la classe de stockage des objets comme suit :

- Lors de la création d'un nouvel objet, vous pouvez spécifier sa classe de stockage. Par exemple, lorsque vous créez des objets à l'aide des opérations d'API PUT Object, POST Object et Initiate Multipart Upload, vous ajoutez l'en-tête de demande `x-amz-storage-class` pour spécifier une classe de stockage. Si vous n'ajoutez pas cet en-tête, Amazon S3 utilise la classe de stockage par défaut, S3 Standard.
- Vous pouvez également modifier la classe de stockage d'un objet déjà stocké dans Amazon S3 vers toute autre classe de stockage en effectuant une copie de l'objet à l'aide de l'opération d'API PUT Object - Copy. Toutefois, vous ne pouvez pas utiliser PUT Object - Copy pour copier des objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Vous ne pouvez pas non plus passer de S3 unizone – Accès peu fréquent à S3 Glacier Instant Retrieval.

  Vous copiez l'objet dans le même compartiment en utilisant le même nom de clé et spécifiez les en-têtes de demande comme suit :

  - Définissez l'en-tête `x-amz-metadata-directive` sur `COPY`.
  - Définissez l'en-tête `x-amz-storage-class` sur la classe de stockage que vous voulez utiliser.

  Dans un compartiment pour lequel la gestion des versions est activée, vous ne pouvez pas modifier la classe de stockage d'une version spécifique d'un objet. Lorsque vous copiez l'objet, Amazon S3 lui attribue un nouvel ID de version.
- Vous pouvez modifier la classe de stockage d'un objet à l'aide de la console Amazon S3 si la taille de l'objet est inférieure à 160 Go. Pour une taille supérieure, nous vous recommandons d'ajouter une configuration de cycle de vie S3 pour changer la classe de stockage de l'objet.
- Si vous utilisez la console Amazon S3 pour modifier la classe de stockage d'un objet doté de balises définies par l'utilisateur, vous devez en avoir l'`s3:GetObjectTagging`autorisation. Si vous modifiez la classe de stockage d'un objet qui ne possède pas de balises définies par l'utilisateur mais dont la taille est supérieure à 16 Mo, vous devez également disposer de l'`s3:GetObjectTagging`autorisation. Si la politique du compartiment de destination refuse

l'`s3:GetObjectTagging`action, la classe de stockage de l'objet sera mise à jour, mais les balises définies par l'utilisateur seront supprimées de l'objet et vous recevrez un message d'erreur.

- Vous pouvez indiquer à Amazon S3 de changer la classe de stockage d'objets en ajoutant la configuration du cycle de vie S3 à un compartiment. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.
- Lors de la définition de la configuration d'une réplication, vous pouvez définir la classe de stockage pour les objets répliqués à toute autre classe de stockage. Cependant, vous ne pouvez pas répliquer des objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Pour plus d'informations, consultez Configuration de réplication.

Restriction des autorisations de stratégie d'accès à une classe de stockage spécifique

Lorsque vous accordez des autorisations de stratégie d'accès pour les opérations Amazon S3, vous pouvez utiliser la clé de condition `s3:x-amz-storage-class` pour restreindre la classe de stockage à utiliser lors du stockage des objets chargés. Par exemple, lorsque vous accordez l'autorisation `s3:PutObject`, vous pouvez restreindre les chargements d'objets à une classe de stockage spécifique. Pour un exemple de politique, consultez Exemple : restriction des téléchargements d'objets vers des objets dotés d'une classe de stockage spécifique.

Pour plus d'informations sur l'utilisation des conditions dans les politiques et une liste complète des clés de condition Amazon S3, consultez les rubriques suivantes :

- Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference
- Exemples de politiques relatives aux compartiments utilisant des clés de condition

Stockage de données à long terme avec les classes de stockage S3 Glacier

Amazon S3 propose plusieurs classes de stockage S3 Glacier conçues pour fournir des solutions économiques pour le stockage à long terme de données peu consultées. Les classes de stockage S3 Glacier sont les suivantes :

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrival
- S3 Glacier Deep Archive

Vous choisissez l'une de ces classes de stockage en fonction de la fréquence à laquelle vous accédez à vos données et de la rapidité avec laquelle vous devez les récupérer. Chacune de ces classes de stockage offre la même durabilité et la même résilience que la classe de stockage S3 Standard, mais à des coûts de stockage inférieurs. Pour plus d'informations sur les classes de stockage S3 Glacier, consultez https://aws.amazon.com/s3/storage-classes/glacier/.

Rubriques

Comparaison des classes de stockage S3 Glacier

Chaque classe de stockage S3 Glacier possède une durée de stockage minimale pour tous les objets. Si vous supprimez, remplacez ou transférez l'objet vers une autre classe de stockage avant le minimum, la durée de stockage minimale complète vous est facturée.

Certaines classes de stockage S3 Glacier sont archivées, ce qui signifie que les objets stockés dans ces classes sont archivés et ne sont pas accessibles en temps réel. Pour plus d’informations, consultez Stockage d'archives.

Les classes de stockage conçues pour des modèles d'accès moins fréquents avec des temps de récupération plus longs permettent de réduire les coûts de stockage. Pour plus d'informations sur les tarifs, consultezhttps://aws.amazon.com/s3/pricing/.

Le tableau suivant récapitule les principaux points à prendre en compte lors du choix d'une classe de stockage S3 Glacier :

S3 Glacier Instant Retrieval

Nous recommandons d'utiliser S3 Glacier Instant Retrieval pour les données à long terme accessibles une fois par trimestre et nécessitant des temps de récupération de quelques millisecondes. Cette classe de stockage est idéale pour les cas d'utilisation sensibles aux

performances tels que l'hébergement d'images, les applications de partage de fichiers et le stockage de dossiers médicaux pour y accéder lors de rendez-vous.

La classe de stockage S3 Glacier Instant Retrieval offre un accès en temps réel à vos objets avec les mêmes performances de latence et de débit que la classe de stockage S3 Standard-IA. Comparé à S3 Standard-IA, S3 Glacier Instant Retrieval présente des coûts de stockage inférieurs mais des coûts d'accès aux données plus élevés.

La taille d'objet minimale est de 128 Ko pour les données stockées dans la classe de stockage S3 Glacier Instant Retrieval. Cette classe de stockage a également une durée de stockage minimale de 90 jours.

S3 Glacier Flexible Retrival

Nous recommandons d'utiliser S3 Glacier Flexible Retrieval pour les données d'archive auxquelles on accède une ou deux fois par an et qui ne nécessitent pas d'accès immédiat. S3 Glacier Flexible Retrieval offre des délais de récupération flexibles pour vous aider à équilibrer les coûts, avec des temps d'accès allant de quelques minutes à quelques heures, et des récupérations groupées gratuites. Cette classe de stockage est idéale pour la sauvegarde et la reprise après sinistre.

Les objets stockés dans S3 Glacier Flexible Retrieval sont archivés et ne sont pas accessibles en temps réel. Pour plus d'informations, consultez Stockage d'archives. Pour accéder à ces objets, vous devez d'abord lancer une demande de restauration qui crée une copie temporaire de l'objet à laquelle vous pouvez accéder une fois la demande terminée. Pour plus d'informations, veuillez consulter Utilisation des objets archivés. Lorsque vous restaurez un objet, vous pouvez choisir un niveau de récupération adapté à votre cas d'utilisation, tout en réduisant les coûts pour des durées de restauration plus longues.

Les niveaux de récupération suivants sont disponibles pour S3 Glacier Flexible Retrieval :

- Récupération accélérée : restaure généralement l'objet en 1 à 5 minutes. Les extractions accélérées sont soumises à la demande. Pour garantir des délais de restauration fiables et prévisibles, nous vous recommandons d'acheter une capacité de récupération provisionnée. Pour plus d'informations, consultez Capacité provisionnée.
- Récupération standard : restaure généralement l'objet en 3 à 5 heures, ou en 1 minute à 5 heures lorsque vous utilisez S3 Batch Operations. Pour plus d'informations, consultez Restauration d'objets à l'aide d'opérations par lot.
- Récupération en masse : restaure généralement l'objet dans un délai de 5 à 12 heures. Les récupérations groupées sont gratuites.

La durée minimale de stockage des objets de la classe de stockage S3 Glacier Flexible Retrieval est de 90 jours.

S3 Glacier Flexible Retrieval nécessite 40 Ko de métadonnées supplémentaires pour chaque objet. Cela inclut 32 Ko de métadonnées nécessaires pour identifier et récupérer vos données, qui sont facturés au tarif par défaut pour S3 Glacier Flexible Retrieval. 8 Ko de données supplémentaires sont nécessaires pour conserver le nom et les métadonnées définis par l'utilisateur pour les objets archivés, et sont facturés au tarif standard S3.

S3 Glacier Deep Archive

Nous recommandons d'utiliser S3 Glacier Deep Archive pour les données d'archive consultées moins d'une fois par an. Cette classe de stockage est conçue pour conserver des ensembles de données pendant plusieurs années afin de répondre aux exigences de conformité et peut également être utilisée pour la sauvegarde ou la reprise après sinistre ou pour toute donnée rarement consultée que vous pouvez attendre jusqu'à 72 heures pour récupérer. S3 Glacier Deep Archive est l'option de stockage la moins chère dans AWS.

Les objets stockés dans S3 Glacier Deep Archive sont archivés et ne sont pas accessibles en temps réel. Pour plus d’informations, consultez Stockage d'archives. Pour accéder à ces objets, vous devez d'abord lancer une demande de restauration qui crée une copie temporaire de l'objet à laquelle vous pouvez accéder une fois la demande terminée. Pour plus d’informations, veuillez consulter Utilisation des objets archivés. Lorsque vous restaurez un objet, vous pouvez choisir un niveau de récupération adapté à votre cas d'utilisation, tout en réduisant les coûts pour des durées de restauration plus longues.

Les niveaux de récupération suivants sont disponibles pour S3 Glacier Deep Archive :

- Récupération standard : restaure généralement l'objet dans les 12 heures, ou dans les 9 à 12 heures lorsque vous utilisez S3 Batch Operations. Pour plus d’informations, consultez Restauration d'objets à l'aide d'opérations par lot.
- Récupération en masse : restaure généralement l'objet dans les 48 heures pour une fraction du coût du niveau de récupération standard.

La durée minimale de stockage des objets de la classe de stockage S3 Glacier Deep Archive est de 180 jours.

S3 Glacier Deep Archive nécessite 40 Ko de métadonnées supplémentaires pour chaque objet. Cela inclut 32 Ko de métadonnées nécessaires pour identifier et récupérer vos données, qui sont facturés

au tarif par défaut pour S3 Glacier Deep Archive. 8 Ko de données supplémentaires sont nécessaires pour conserver le nom et les métadonnées définis par l'utilisateur pour les objets archivés, et sont facturés au tarif standard S3.

Stockage d'archives

S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont des classes de stockage d'archives. Cela signifie que lorsque vous stockez un objet dans ces classes de stockage, cet objet est archivé et n'est pas directement accessible. Pour accéder à un objet archivé, vous soumettez une demande de restauration, puis vous attendez que le service restaure l'objet. La demande de restauration restaure une copie temporaire de l'objet, qui est supprimée lorsque la durée spécifiée dans la demande expire. Pour plus d'informations, consultez Utilisation des objets archivés.

Ces classes de stockage nécessitent 40 Ko de métadonnées supplémentaires pour chaque objet archivé. Cela inclut 32 Ko de métadonnées nécessaires pour identifier et récupérer vos données, qui sont facturés au tarif par défaut pour cette classe de stockage. 8 Ko de données supplémentaires sont nécessaires pour conserver le nom et les métadonnées définis par l'utilisateur pour les objets archivés, et sont facturés au tarif standard S3.

Les objets appartenant à ces classes de stockage sont facturés aux tarifs de la classe de stockage S3 Standard lorsque vous les chargez à l'aide de téléchargements partitionnés. Pour plus d'informations, consultez Chargement partitionné et tarification.

Vous pouvez restaurer des objets archivés dans ces classes de stockage avec un maximum de 1 000 transactions par seconde (TPS) de demandes de restauration d'objets par compte et par compte. Région AWS

En quoi ces classes de stockage diffèrent-elles du service S3 Glacier

Les classes de stockage S3 Glacier font partie du service Amazon S3 et stockent les données sous forme d'objets dans des compartiments S3. Vous pouvez gérer les objets de ces classes de stockage à l'aide de la console S3 ou par programmation à l'aide des API ou SDK S3. Lorsque vous stockez des objets dans des classes de stockage S3 Glacier, vous pouvez utiliser les fonctionnalités S3 telles que le chiffrement avancé, le balisage des objets et les configurations du cycle de vie S3 pour vous aider à gérer l'accessibilité et les coûts des données.

> **Important**
>
> Nous vous recommandons d'utiliser les classes de stockage S3 Glacier au sein du service Amazon S3 pour toutes vos données à long terme.

Le service Amazon S3 Glacier (S3 Glacier) est un service distinct qui stocke les données sous forme d'archives dans des coffres-forts. Ce service ne prend pas en charge les fonctionnalités d'Amazon S3 et ne fournit pas de support de console pour les opérations de chargement et de téléchargement de données. Nous vous déconseillons d'utiliser le service S3 Glacier pour vos données à long terme. Les données stockées dans ce service ne sont pas accessibles depuis le service Amazon S3. Si vous recherchez des informations sur le service S3 Glacier, consultez le guide du développeur Amazon S3 Glacier. Pour transférer des données du service Amazon S3 Glacier vers une classe de stockage dans Amazon S3, voir Transfert de données d'Amazon S3 Glacier Vaults vers Amazon S3 dans la bibliothèque de AWS solutions.

Amazon S3 Intelligent Tiering

La classe de stockage S3 Intelligent-Tiering est conçue pour optimiser les coûts de stockage en déplaçant automatiquement les données vers le niveau d'accès de stockage le plus rentable lorsque les modèles d'accès changent, sans incidence sur les performances ni sur les frais d'exploitation. Moyennant des frais mensuels minimes de surveillance et d'automatisation des objets, S3 Intelligent-Tiering surveille les modèles d'accès et déplace automatiquement les objets qui n'ont pas été consultés vers des niveaux d'accès moins coûteux.

S3 Intelligent-Tiering permet de réduire automatiquement les coûts de stockage grâce à trois niveaux d'accès à faible latence et à haut débit. Pour les données accessibles de manière asynchrone, vous pouvez choisir d'activer les capacités d'archivage automatique dans la classe de stockage S3 Intelligent-Tiering. L'utilisation de S3 Intelligent-Tiering n'entraîne aucuns frais de récupération. Si un objet du niveau d'accès Peu fréquent ou Archive Instant est consulté ultérieurement, il est automatiquement replacé dans le niveau d'accès Fréquent. Aucuns frais supplémentaires ne s'appliquent lorsque des objets sont déplacés entre des niveaux d'accès de la classe de stockage S3 Intelligent-Tiering.

S3 Intelligent-Tiering est la classe de stockage recommandée pour les données avec des modèles d'accès inconnus, changeants ou imprévisibles, indépendamment de la taille de l'objet ou de la période de rétention, telles que les lacs de données, l'analytique des données et les nouvelles applications.

Pour en savoir sur l'utilisation de S3 Intelligent-Tiering, consultez les sections suivantes :

Rubriques

Fonctionnement de S3 Intelligent-Tiering

La classe de stockage Amazon S3 Intelligent-Tiering stocke automatiquement les objets dans trois niveaux d'accès. Un niveau est optimisé pour les accès fréquents, un niveau à coût réduit est optimisé pour les accès peu fréquents, et un autre niveau au coût très réduit est optimisé pour les données rarement accessibles. Pour un faible coût mensuel de surveillance et d'automatisation des objets, S3 Intelligent-Tiering surveille les modèles d'accès et déplace automatiquement les objets vers le niveau d'accès Peu fréquent lorsque ceux-ci n'ont pas été consultés pendant 30 jours consécutifs. Après 90 jours sans accès, les objets sont déplacés vers le niveau d'accès Archive Instant sans impact sur les performances ni sur les frais d'exploitation.

Pour obtenir le coût de stockage le plus bas sur des données auxquelles on peut accéder en quelques minutes ou quelques heures, activez les capacités d'archivage pour ajouter deux niveaux d'accès supplémentaires. Vous pouvez répartir les objets sur le niveau d'accès Archive, le niveau d'accès Deep Archive, ou les deux. Avec l'accès Archive, S3 Intelligent-Tiering déplace les objets qui n'ont pas été consultés pendant un minimum de 90 jours consécutifs vers le niveau d'accès Archive. Avec l'accès Deep Archive, S3 Intelligent-Tiering déplace les objets vers le niveau d'accès Deep Archive après un minimum de 180 jours consécutifs sans accès. Pour les deux niveaux, vous pouvez configurer le nombre de jours sans accès en fonction de vos besoins.

Les actions suivantes constituent un accès qui empêche de classer vos objets dans le niveau d'accès Archive ou le niveau d'accès Deep Archive :

- Télécharger ou copier un objet archivé via la console Amazon S3.
- Invocation de CopyObject, UploadPartCopy ou réplication d'objets avec la réplication par lot S3. Dans ces cas, les objets source des opérations de copie ou de réplication sont hiérarchisés.
- Invocation de GetObject, PutObject, RestoreObject, CompleteMultipartUpload, ListParts ou SelectObjectContent.

Par exemple, si vous accédez à vos objets via `SelectObjectContent` avant le nombre de jours sans accès que vous avez spécifié (180 jours, par exemple), cette action réinitialise le chronomètre. Vos objets ne sont pas déplacés vers le niveau d'accès Archive ni vers le niveau d'accès Deep Archive tant que la dernière demande `SelectObjectContent` n'a pas atteint le nombre de jours spécifiés.

Si un objet du niveau d'accès Peu fréquent ou Archive Instant est consulté ultérieurement, il est automatiquement replacé dans le niveau d'accès Fréquent.

Les actions suivantes constituent un accès qui renvoie automatiquement des objets du niveau d'accès Peu fréquent ou Archive Instant vers le niveau d'accès Fréquent :

- Télécharger ou copier un objet archivé via la console Amazon S3.
- Invocation de CopyObject, UploadPartCopy ou réplication d'objets avec la réplication par lot. Dans ces cas, les objets source des opérations de copie ou de réplication sont hiérarchisés.
- Invocation de GetObject, PutObject, RestoreObject, CompleteMultipartUpload ou ListParts.

Les autres actions ne constituent pas un accès qui renvoie automatiquement des objets du niveau d'accès Peu fréquent ou Archive Instant vers le niveau d'accès Fréquent. Voici un exemple, et non une liste définitive, de telles actions :

- Invocation de HeadObject, GetObjectTagging, PutObjectTagging, ListObjects, ListObjectsV2 ou ListObjectVersions.
- L'invocation de SelectObjectContent ne constitue pas un accès qui fait passer les objets au niveau d'accès Fréquent. De plus, elle n'empêche pas de hiérarchiser les objets du niveau d'accès fréquent au niveau d'accès peu fréquent, puis jusqu'au niveau Archive Instant Access.

Vous pouvez configurer S3 Intelligent-Tiering comme votre classe de stockage par défaut pour les données récemment créées en indiquant `INTELLIGENT-TIERING` dans votre en-tête de demande PutBucketIntelligentTieringConfiguration. S3 Intelligent-Tiering est conçue pour une disponibilité de 99,9 % et une durabilité de 99,9999999 %.

> **Note**
>
> Si la taille d'un objet est inférieure à 128 Ko, il n'est pas contrôlé et n'est pas admissible à la hiérarchisation automatique. Les objets plus petits sont toujours stockés dans le niveau Accès fréquent.

Niveaux d'accès S3 Intelligent-Tiering.

La section suivante explique les différents niveaux d'accès automatiques et facultatifs. Lorsque des objets se déplacent entre les niveaux d'accès, la classe de stockage reste la même (S3 Intelligent-Tiering).

Niveau Accès fréquent (automatique)

C'est le niveau d'accès par défaut dans lequel tout objet créé ou migré dans S3 Intelligent-Tiering commence son cycle de vie. Un objet reste dans ce niveau tant qu'il est accessible. Le niveau d'accès fréquent offre une faible latence et des performances de débit élevées.

Niveau d'accès peu fréquent (automatique)

Si vous n'accédez pas à un objet pendant 30 jours consécutifs, il passera au niveau Accès peu fréquent. Le niveau d'accès peu fréquent offre une faible latence et des performances de débit élevées.

Niveau d'accès Archive Instant (automatique)

Si vous n'accédez pas à un objet pendant 90 jours consécutifs, il passera au niveau d'accès Archive Instant. Le niveau Archive Instant Access offre une faible latence et des performances de débit élevées.

Niveau d'accès Archive (facultatif)

S3 Intelligent-Tiering vous offre la possibilité d'activer le niveau d'accès Archive pour les données accessibles de manière asynchrone. Une fois activé, le niveau d'accès Archive archive automatiquement les objets qui n'ont pas été consultés pendant un minimum de 90 jours consécutifs. Vous pouvez prolonger le délai de dernier accès pour l'archivage jusqu'à un maximum de 730 jours. Le niveau d'accès Archive offre les mêmes performances que la classe de stockage S3 Glacier Flexible Retrieval.

Les temps de récupération standard pour ce niveau d'accès peuvent varier de 3 à 5 heures. Si vous lancez votre demande de restauration à l'aide d'opérations par lot S3, votre restauration démarre en quelques minutes. Pour plus d'informations sur les options et les durées de récupération, consultez the section called “Restauration des objets à partir des niveaux d'accès Archive et Deep Archive de S3 Intelligent-Tiering”.

> **Note**
>
> N'activez le niveau d'accès Archive que pendant 90 jours si vous souhaitez contourner le niveau d'accès Archive Instant. Le niveau Archive Access offre des coûts de stockage légèrement inférieurs avec des durées de récupération allant de quelques minutes à plusieurs heures. Le niveau d'accès Archive Instant offre un accès de l'ordre de la milliseconde et des performances à haut débit.

Niveau d'accès Deep Archive (facultatif)

S3 Intelligent-Tiering vous offre la possibilité d'activer le niveau d'accès Deep Archive pour les données accessibles de manière asynchrone. Une fois activé, le niveau d'accès Deep Archive archive automatiquement les objets qui n'ont pas été consultés pendant un minimum de 180 jours consécutifs. Vous pouvez prolonger le délai de dernier accès pour l'archivage jusqu'à un maximum de 730 jours. Le niveau d'accès Deep Archive offre les mêmes performances que la classe de stockage S3 Glacier Deep Archive.

La récupération standard des objets dans ce niveau d'accès a lieu dans les 12 heures. Si vous lancez votre demande de restauration à l'aide d'opérations par lot S3, votre restauration démarre dans les 9 heures. Pour plus d'informations sur les options et les durées de récupération, consultez the section called “Restauration des objets à partir des niveaux d'accès Archive et Deep Archive de S3 Intelligent-Tiering”.

> **Note**
>
> Activez les niveaux d'accès Archive et Deep Archive uniquement si vos objets sont accessibles de manière asynchrone par votre application. Si l'objet que vous récupérez est stocké au niveau d'accès Archive ou Deep Archive, vous devez d'abord restaurer l'objet à l'aide de l'opération `RestoreObject`.

Utiliser S3 Intelligent-Tiering

Vous pouvez utiliser la classe de stockage S3 Intelligent-Tiering afin d'optimiser automatiquement les coûts de stockage. S3 Intelligent-Tiering permet de réaliser automatiquement des économies en déplaçant les données à un niveau d'objet granulaire entre les niveaux d'accès lorsque les modèles

d'accès changent. Pour les données accessibles de manière asynchrone, vous pouvez choisir d'activer l'archivage automatique dans la classe de stockage S3 Intelligent-Tiering à l'aide de AWS Management Console, AWS CLI ou API Amazon S3.

Déplacement des données vers S3 Intelligent-Tiering

Il existe deux façons de déplacer des données vers S3 Intelligent-Tiering. Vous pouvez PUT (mettre) directement des données dans S3 Intelligent-Tiering en spécifiant `INTELLIGENT_TIERING` dans l'en-tête `x-amz-storage-class` ou configurer les configurations de cycle de vie S3 pour migrer les objets depuis S3 Standard ou S3 Standard-Accès peu fréquent vers S3 Intelligent-Tiering.

Chargement de données vers S3 Intelligent-Tiering à l'aide de Direct PUT

Lorsque vous chargez un objet vers la classe de stockage S3 Intelligent-Tiering à l'aide de l'opération d'API PUT, vous spécifiez S3 Intelligent-Tiering dans l'en-tête de demande `x-amz-storage-class`.

La requête suivante stocke l'image, `my-image.jpg`, dans le `myBucket`compartiment. La requête utilise l'en-tête `x-amz-storage-class` pour demander que l'objet soit stocké à l'aide de la classe de stockage S3 Intelligent-Tiering.

Example

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.<Region>.amazonaws.com (http://amazonaws.com/)
Date: Wed, 1 Sep 2021 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Transition des données vers S3 Intelligent-Tiering depuis S3 standard ou S3 Standard-Accès peu fréquent à l'aide du cycle de vie S3

Vous pouvez ajouter des règles à une configuration de cycle de vie S3 afin d'indiquer à Amazon S3 d'effectuer la transition des objets d'une classe de stockage vers une autre. Pour plus d'informations sur les transitions prises en charge et les contraintes associées, consultez Transition des objets à l'aide du cycle de vie S3.

Vous pouvez spécifier des configurations de cycle de vie S3 au niveau du compartiment ou du préfixe. Dans cette règle de configuration de cycle de vie S3, le filtre spécifie un préfixe de clé

(`documents/`). Par conséquent, la règle s'applique aux objets avec le préfixe du nom de clé `documents/`, comme `documents/doc1.txt` et `documents/doc2.txt` La règle spécifie une action `Transition` indiquant à Amazon S3 d'effectuer la transition d'objets vers la classe de stockage S3 Intelligent-Tiering 0 jour après leur création. Dans ce cas, les objets sont admissibles à la transition vers S3 Intelligent-Tiering à minuit TUC après leur création.

Example

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
       <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>INTELLIGENT_TIERING</StorageClass>
    </Transition>
 </Rule>
</LifecycleConfiguration>
```

Activation des niveaux d'accès S3 Intelligent-Tiering Archive et Deep Archive

Pour bénéficier des coûts de stockage les plus bas pour les données accessibles sur une période allant de quelques minutes à plusieurs heures, vous pouvez activer l'un ou les deux niveaux d'accès d'archivage en créant une configuration de niveau du compartiment, du préfixe ou de la balise d'objet à l'aide de AWS Management Console, AWS CLI ou de l'API Amazon S3.

Utilisation de la console S3

Activer l'archivage automatique S3 Intelligent-Tiering

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom du compartiment que vous souhaitez utiliser.
3. Choisissez Propriétés.
4. Accédez à la section Configuration d'archivage S3 Intelligent-Tiering, puis choisissez Créer une configuration.

5. Dans la section Paramètres de configuration d'archivage, précisez un nom de configuration descriptif pour votre configuration d'archivage S3 Intelligent-Tiering.
6. Sous Choisir une portée de configuration, choisissez une portée de configuration à utiliser. Le cas échéant, vous pouvez limiter la portée de configuration aux objets spécifiés dans un compartiment à l'aide d'un préfixe partagé, d'une balise d'objet ou d'une combinaison des deux.
   a. Pour limiter la portée de la configuration, sélectionnez Limiter la portée de cette configuration à l'aide d'un ou de plusieurs filtres.
   b. Pour limiter la portée de la configuration à l'aide d'un préfixe unique, saisissez le préfixe sous Préfixe.
   c. Pour limiter la portée de la configuration à l'aide de balises d'objet, sélectionnez Ajouter une balise, puis saisissez une valeur de clé.
7. Sous État, sélectionnez Activer.
8. Dans la section Paramètres Archive, sélectionnez l'un ou les deux niveaux d'accès Archive à activer.
9. Choisissez Créer.

Utilisation de AWS CLI

Pour gérer les configurations S3 Intelligent-Tiering, vous pouvez utiliser les commandes AWS CLIsuivantes :

- `delete-bucket-intelligent-tiering-configuration`
- `get-bucket-intelligent-tiering-configuration`
- `list-bucket-intelligent-tiering-configurations`
- `put-bucket-intelligent-tiering-configuration`

Pour savoir comment configurer AWS CLI, consultez Développement avec Amazon S3 à l'aide de la AWS CLI.

Lorsque vous utilisez AWS CLI, vous ne pouvez pas spécifier la configuration en tant que fichier XML. Vous devez spécifier le format JSON à la place. Voici un exemple de configuration S3 Intelligent-Tiering au format XML et de son équivalent au format JSON que vous pouvez spécifier dans une commande AWS CLI.

L'exemple suivant montre comment placer une configuration S3 Intelligent-Tiering dans le compartiment spécifié.

Example **put-bucket-intelligent-tiering-configuration**

JSON

```
{
  "Id": "string",
  "Filter": {
    "Prefix": "string",
    "Tag": {
      "Key": "string",
      "Value": "string"
    },
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
        ...
      ]
    }
  },
  "Status": "Enabled"|"Disabled",
  "Tierings": [
    {
      "Days": integer,
      "AccessTier": "ARCHIVE_ACCESS"|"DEEP_ARCHIVE_ACCESS"
    }
    ...
  ]
}
```

XML

```
PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Id>string</Id>
```

```
   <Filter>
      <And>
         <Prefix>string</Prefix>
         <Tag>
            <Key>string</Key>
            <Value>string</Value>
         </Tag>
         ...
      </And>
      <Prefix>string</Prefix>
      <Tag>
         <Key>string</Key>
         <Value>string</Value>
      </Tag>
   </Filter>
   <Status>string</Status>
   <Tiering>
      <AccessTier>string</AccessTier>
      <Days>integer</Days>
   </Tiering>
   ...
</IntelligentTieringConfiguration>
```

Utilisation de l'opération d'API PUT

Vous pouvez utiliser l'opération `PutBucketIntelligentTieringConfiguration` pour un compartiment spécifié et jusqu'à 1 000 configurations S3 Intelligent-Tiering par compartiment. Vous pouvez définir les objets d'un compartiment qui peuvent bénéficier des niveaux d'accès Archive à l'aide d'un préfixe partagé ou d'une balise d'objet. Utiliser un préfixe partagé ou une balise d'objet permet de faire correspondre des applications métier, des flux de travail ou des organisations internes spécifiques. Vous avez également la possibilité d'activer le niveau d'accès Archive, le niveau d'accès Deep Archive, ou les deux.

Démarrer avec Amazon S3 Intelligent-Tiering

Pour en savoir plus sur l'utilisation de S3 Intelligent-Tiering, consultez Tutorial: Getting started using S3 Intelligent-Tiering (Tutoriel : démarrer avec Amazon S3 Intelligent-Tiering).

Gestion de S3 Intelligent-Tiering

Le classe de stockage S3 Intelligent-Tiering permet de réduire automatiquement les coûts de stockage grâce à trois niveaux d'accès à faible latence et à haut débit. Il offre également des fonctionnalités d'archivage en option vous permettant de bénéficier des coûts de stockage les plus bas dans le cloud pour les données accessibles sur une période allant de quelques minutes à plusieurs heures. La classe de stockage S3 Intelligent-Tiering prend en charge toutes les fonctions Amazon S3, y compris les suivantes :

- S3 Inventory, pour vérifier le niveau d'accès des objets
- S3 Replication, pour répliquer des données vers n'importe quel Région AWS
- S3 Storage Lens, pour afficher les métriques d'utilisation et d'activité du stockage
- Chiffrement côté serveur, pour protéger les données objet
- Verrouillage des objets S3, pour empêcher la suppression accidentelle de données
- AWS PrivateLink, pour accéder à Amazon S3 via un point de terminaison privé dans un cloud privé virtuel (VPC)

Identification des objets du niveau d'accès S3 Intelligent-Tiering qui sont stockés dans

Pour obtenir une liste de vos objets et de leurs métadonnées correspondantes, y compris leur niveau d'accès S3 Intelligent-Tiering, vous pouvez utiliser. the section called “Gestion de l'inventaire” L'inventaire S3 fournit des fichiers de sortie CSV, ORC ou Parquet qui répertorient vos objets et leurs métadonnées correspondantes. Vous pouvez recevoir ces rapports d'inventaire sur une base quotidienne ou hebdomadaire pour un compartiment Amazon S3 ou un préfixe partagé. (Le préfixe partagé fait référence aux objets dont les noms commencent par une chaîne commune.)

Affichage de l'état de l'archive d'un objet dans S3 Intelligent-Tiering

Vous pouvez configurer des notifications d'événements S3 afin d'être informé lorsqu'un objet de la classe de stockage S3 Intelligent-Tiering passe au niveau Archive Access ou Deep Archive Access. Pour en savoir plus, consultez Activation des notifications d'événement.

Amazon S3 peut publier des notifications d'événement dans une rubrique Amazon Simple Notification Service (Amazon SNS), une file d'attente Amazon Simple Queue Service (Amazon SQS) ou une fonction AWS Lambda . Pour plus d’informations, consultez Notifications d'événements Amazon S3.

Le message suivant est un exemple de message envoyé par Amazon S3 pour publier un événement `s3:IntelligentTiering`. Pour plus d'informations, consultez the section called "Structure des messages d'événements".

```
{
   "Records":[
      {
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"IntelligentTiering",
         "userIdentity":{
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"mybucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::mybucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
            }
         },
         "intelligentTieringEventData":{
            "destinationAccessTier": "ARCHIVE_ACCESS"
            }
      }
```

```
    ]
}
```

Vous pouvez également utiliser une demande d'objet HEAD pour afficher le statut d'archivage d'un objet. Si un objet est stocké dans la classe de stockage S3 Intelligent-Tiering et se trouve dans l'un des niveaux d'archivage, la réponse de l'objet HEAD affiche le niveau d'archivage actuel. Pour afficher le niveau d'archivage, la demande utilise l'en-tête x-amz-archive-status.

La demande d'objet HEAD suivante renvoie les métadonnées d'un objet (dans ce cas, *my-image.jpg*).

Example

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.region.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

Vous pouvez également utiliser des demandes d'objet HEAD pour surveiller le statut d'une demande `restore-object`. Si la restauration de l'archive est en cours, la réponse de l'objet HEAD inclut l'en-tête x-amz-restore.

L'exemple suivant de réponse d'objet HEAD montre un objet archivé à l'aide de S3 Intelligent-Tiering avec une demande de restauration en cours.

Example

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeu7M19iI8UbxMbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
x-amz-storage-class: 'INTELLIGENT_TIERING'
x-amz-archive-status: 'ARCHIVE_ACCESS'
x-amz-restore: 'ongoing-request="true"'
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'
Accept-Ranges: bytes
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Restauration des objets à partir des niveaux d'accès Archive et Deep Archive de S3 Intelligent-Tiering

Pour accéder aux objets des niveaux S3 Intelligent-Tiering Archive Access et Deep Archive Access, vous devez lancer une demande de restauration, puis attendre que l'objet soit déplacé vers le niveau Frequent Access. Pour plus d'informations sur les objets archivés, consultezthe section called “Utilisation des objets archivés”.

Lorsque vous restaurez un objet depuis les niveaux d'accès Archive et Deep Archive, l'objet retourne au niveau d'accès Fréquent. Par la suite, si vous n'accédez pas à l'objet dans les 30 jours consécutifs, il sera automatiquement déplacé vers le niveau d'accès Peu fréquent. Ensuite, après un minimum de 90 jours consécutifs sans accès, l'objet passe au niveau Archive Access. Après un minimum de 180 jours consécutifs sans accès, l'objet passe au niveau Deep Archive Access. Pour plus d’informations, consultez the section called “Fonctionnement de S3 Intelligent-Tiering”.

Vous pouvez restaurer un objet archivé à l'aide de la console Amazon S3, de S3 Batch Operations, de l'API REST Amazon S3, AWS des SDK ou du AWS Command Line Interface (AWS CLI). Pour plus d'informations, voir the section called “Utilisation des objets archivés”.

Gestion du cycle de vie de votre stockage

Pour gérer vos objets afin qu'ils soient stockés de manière rentable tout au long de leur cycle de vie, créez une configuration Amazon S3 Lifecycle. Une configuration du cycle de vie Amazon S3 est un ensemble de règles qui définissent les actions qu'Amazon S3 applique à un groupe d'objets. Il existe deux types d'actions :

- Actions de transition : ces actions définissent à quel moment les objets effectuent la transition vers une autre classe de stockage. Par exemple, vous pouvez choisir d'effectuer la transition des objets vers la classe de stockage S3 standard – Accès peu fréquent 30 jours après leur création ou de les archiver dans la classe de stockage S3 Glacier Flexible Retrieval un an après leur création. Pour plus d’informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

  Des coûts sont associés aux demandes de transition de cycle de vie. Pour en savoir plus sur la tarification, veuillez consulter Tarification Simple Storage Service (Amazon S3).
- Actions d'expiration : ces actions définissent la date d'expiration des objets. Amazon S3 supprime les objets expirés à votre place.

Les coûts d'expiration de cycle de vie dépendent du moment où vous choisissez de faire expirer des objets. Pour plus d'informations, consultez Objets en cours d'expiration.

> **Important**
>
> Vous ne pouvez pas utiliser une politique de compartiment pour empêcher les suppressions ou les transitions selon une règle du cycle de vie S3. Par exemple, même si votre politique de compartiment refuse toutes les actions pour tous les principaux, votre configuration S3 Lifecycle fonctionne toujours normalement.

Objets existants et nouveaux

Quand vous ajoutez une configuration de cycle de vie dans un compartiment, les règles de configuration s'appliquent à la fois aux objets existants et à ceux que vous ajouterez ultérieurement. Par exemple, si vous ajoutez une règle de configuration du cycle de vie aujourd'hui avec une action d'expiration qui entraîne l'expiration des objets 30 jours après leur création, Amazon S3 mettra en file d'attente pour suppression tous les objets existants vieux de plus de 30 jours.

Changements au niveau de la facturation

En cas de retard entre le moment où un objet devient éligible à une action de cycle de vie et le moment où Amazon S3 transfère ou fait expirer votre objet, les modifications de facturation sont appliquées dès que l'objet devient éligible à l'action de cycle de vie. Par exemple, si l'expiration d'un objet est programmée et qu'Amazon S3 ne l'expire pas immédiatement, le stockage ne vous sera pas facturé après le délai d'expiration.

Il existe toutefois une exception à ce comportement si vous disposez d'une règle de cycle de vie configurée pour transférer l'objet à la classe de stockage S3 Intelligent-Tiering. Dans ce cas, les changements de facturation ne se produisent pas tant que l'objet n'est pas passé à la classe de stockage S3 Intelligent-Tiering.

Pour plus d'informations sur les règles de cycle de vie S3, consultez Éléments de la configuration du cycle de vie.

Surveillance de l'effet des règles relatives au cycle de vie

Pour surveiller l'effet des mises à jour effectuées par les règles de cycle de vie actives, voirthe section called “Comment puis-je contrôler les mesures prises conformément à mes règles de cycle de vie ?”.

Gestion du cycle de vie des objets

Définissez des règles de configuration du cycle de vie S3 pour des objets ayant un cycle de vie bien défini. Exemples :

- Si vous chargez régulièrement des journaux dans votre compartiment, il se peut que l'application ait besoin de ces journaux pendant une semaine ou un mois. À l'issue de cette période, vous pouvez souhaiter les supprimer.
- Certains documents sont consultés fréquemment pendant une période limitée. À l'issue de cette période, ils sont moins souvent consultés. À un moment donné, il est possible que vous n'ayez pas besoin d'y accéder en temps réel, mais votre entreprise ou la législation peut vous imposer de les archiver pour une durée spécifique. À l'issue de cette période, vous pouvez les supprimer.
- Vous pouvez télécharger certains types de données sur Amazon S3, essentiellement à des fins d'archivage. Par exemple, vous pouvez archiver des multimédias numériques, des enregistrements financiers et des enregistrements liés à la santé, des données brutes sur la séquence du génome, des sauvegardes de bases de données à long terme et des données à conserver pour des raisons de conformité réglementaire.

Avec des règles de configuration de cycle de vie S3, vous pouvez indiquer à Amazon S3 d'effectuer la transition des objets vers des classes de stockage moins onéreuses, de les archiver ou de les supprimer.

Création d’une configuration de cycle de vie

Une configuration de cycle de vie S3 est un fichier XML qui comprend un ensemble de règles avec des actions prédéfinies que vous souhaitez que Amazon S3 exécute sur des objets durant leur cycle de vie.

Vous pouvez créer une configuration du cycle de vie à l'aide de la console Amazon S3, de l'API REST, AWS des SDK et du AWS Command Line Interface (AWS CLI). Pour plus d’informations, consultez Configuration du cycle de vie d'un bucket.

Amazon S3 fournit un ensemble d'opérations d'API REST pour gérer la configuration de cycle de vie sur un compartiment. Amazon S3 stocke la configuration en tant que sous-ressource du cycle de vie attachée à votre compartiment. Pour plus d'informations, consultez la:

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Pour en savoir plus sur la configuration du cycle de vie, consultez les rubriques suivantes :

Rubriques

- Transition des objets à l'aide du cycle de vie Amazon S3
- Objets en cours d'expiration
- Configuration du cycle de vie d'un bucket
- Cycle de vie et autres configurations de compartiment
- Configuration des notifications d'événements de cycle de vie
- Éléments de la configuration du cycle de vie
- Exemples de configuration de cycle de vie S3

Transition des objets à l'aide du cycle de vie Amazon S3

Vous pouvez ajouter des règles à une configuration de cycle de vie S3 afin d'indiquer à Amazon S3 d'effectuer la transition des objets vers une autre classe de stockage Amazon S3. Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). Voici quelques exemples de situations où vous pouvez utiliser les configurations du cycle de vie S3 de cette manière :

- Lorsque vous savez qu'il s'agit d'objets à accès peu fréquent, vous pouvez effectuer leur transition vers la classe de stockage S3 standard – Accès peu fréquent.
- Vous souhaiterez peut-être archiver des objets auxquels vous n'avez pas besoin d'accéder en temps réel dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

Objets existants et nouveaux

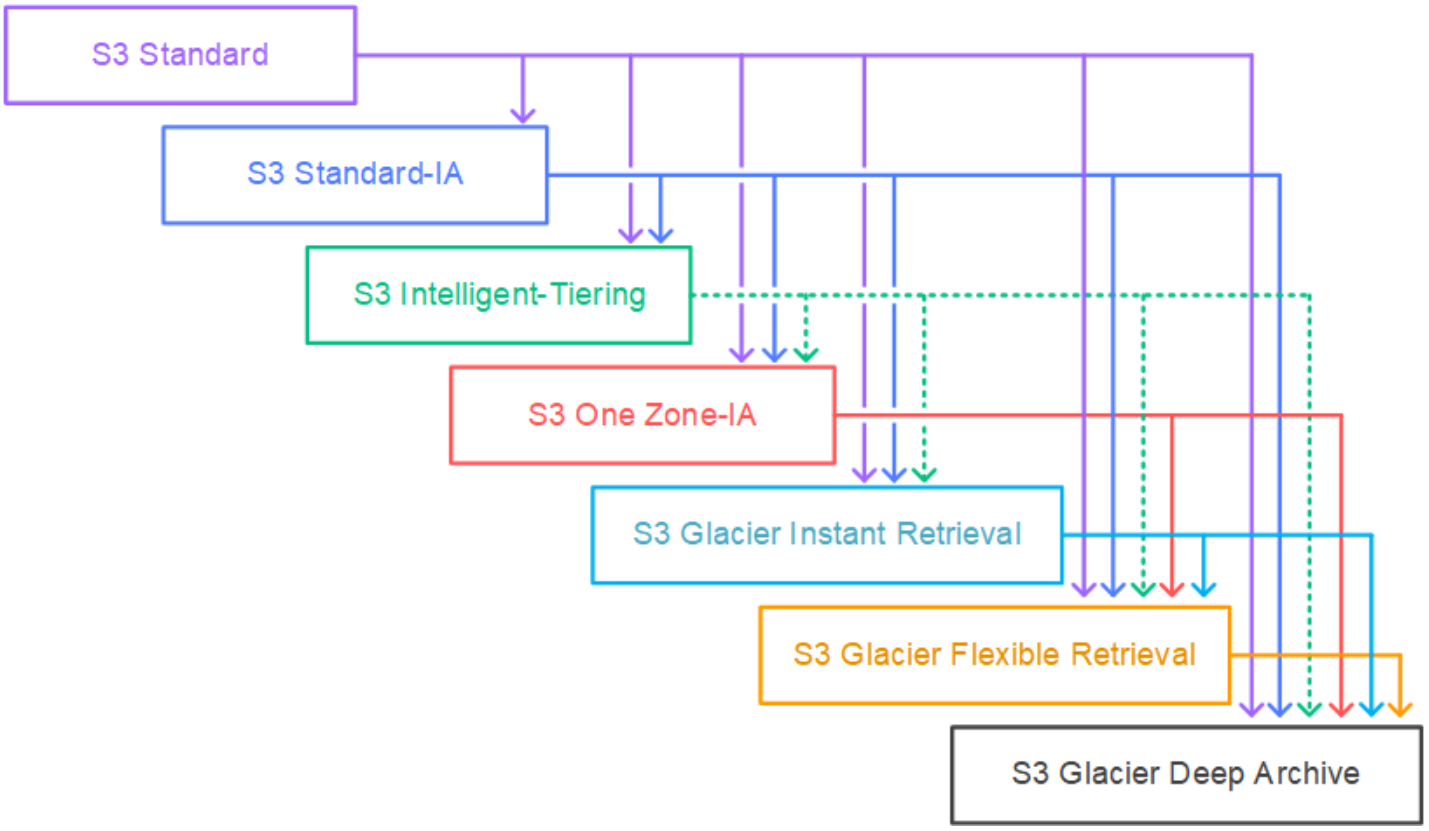
Quand vous ajoutez une configuration de cycle de vie dans un compartiment, les règles de configuration s'appliquent à la fois aux objets existants et à ceux que vous ajouterez ultérieurement. Par exemple, si vous ajoutez une règle de configuration du cycle de vie aujourd'hui avec une action de transition qui permet aux objets dotés d'un préfixe spécifique de passer à une autre classe de stockage 30 jours après leur création, Amazon S3 mettra en file d'attente pour la transition tous les objets existants âgés de plus de 30 jours et portant le préfixe spécifié.

> **Important**
>
> Vous ne pouvez pas utiliser une politique de compartiment pour empêcher les suppressions ou les transitions selon une règle du cycle de vie S3. Par exemple, même si votre politique de compartiment refuse toutes les actions pour tous les principaux, votre configuration S3 Lifecycle fonctionne toujours normalement.

Transitions prises en charge et contraintes connexes

Dans une configuration de cycle de vie S3, vous pouvez définir des règles afin d'effectuer la transition d'objets d'une classe de stockage vers une autre et économiser les coûts de stockage. Si vous ne connaissez pas les modèles d'accès de vos objets ou que vos modèles d'accès évoluent au fil du temps, vous pouvez effectuer la transition des objets vers la classe de stockage S3 Intelligent-Tiering pour réduire automatiquement les coûts. Pour obtenir des informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Amazon S3 prend en charge un modèle en cascade pour la transition entre classes de stockage, comme illustré dans le schéma suivant.



Transitions de cycle de vie prises en charge

Amazon S3 prend en charge les transitions de cycle de vie suivantes entre les classes de stockage à l'aide d'une configuration de cycle de vie S3.

Vous pouvez effectuer une transition entre les classes des façons suivantes :

- La classe de stockage S3 standard vers n'importe quelle autre classe de stockage.
- La classe de stockage S3 standard – Accès peu fréquent vers les classes de stockage S3 Intelligent-Tiering, S3 unizone – Accès peu fréquent ou S3 Glacier Instant Retrieval.
- La classe de stockage S3 Intelligent-Tiering vers les classes de stockage S3 unizone – Accès peu courant, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

> **Note**
>
> Certaines exceptions concernent la transition d'objets de la classe de stockage S3 Intelligent-Tiering vers S3 One Zone-IA et certaines classes de stockage S3 Glacier. Pour plus d'informations, consultez the section called “Transitions de cycle de vie non prises en charge”.

- La classe de stockage S3 unizone – Accès peu fréquent vers les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- La classe de stockage S3 Glacier Instant Retrieval vers les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.
- La classe de stockage S3 Glacier Flexible Retrieval vers la classe de stockage S3 Glacier Deep Archive.
- Toute classe de stockage vers la classe de stockage S3 Glacier Deep Archive.

> **Note**
>
> Aucuns frais de récupération de données ne sont facturés pour les transitions de cycle de vie. Cependant, des frais d'ingestion par demande sont facturés lors de l'utilisation PUT ou des règles de cycle de vie pour déplacer des données vers n'importe quelle classe de stockage S3 sont applicables. COPY Tenez compte du coût d'ingestion ou de transition avant de déplacer des objets dans n'importe quelle classe de stockage. Pour plus d'informations sur les coûts, consultez Tarification Amazon S3.

Transitions de cycle de vie non prises en charge

Amazon S3 ne prend en charge aucune des transitions de cycle de vie suivantes.

Vous ne pouvez pas effectuer la transition entre les classes suivantes :

- Toute classe de stockage vers la classe de stockage S3 standard.
- Toute classe de stockage vers la classe de stockage à redondance réduite (RRS).
- La classe de stockage S3 unizone – Accès peu fréquent vers les classes de stockage S3 Intelligent-Tiering, S3 standard – Accès peu fréquent ou S3 Glacier Instant Retrieval.
- De la classe de stockage S3 Intelligent-Tiering (tous les niveaux) à la classe de stockage S3 Standard-IA.
- La classe de stockage S3 Intelligent-Tiering Archive Instant Access passe au niveau S3 One Zone-IA.
- La classe de stockage S3 Intelligent-Tiering permet d'accéder aux archives au niveau S3 One Zone-IA ou S3 Glacier Instant Retrieval.
- La classe de stockage S3 Intelligent-Tiering Deep Archive Access passe à S3 One Zone-IA, S3 Glacier Instant Retrieval ou S3 Glacier Flexible Retrieval.

Constraints

Les transitions des classes de stockage de cycle de vie obéissent aux contraintes suivantes :

Taille de l’objet et transitions de la classe S3 standard ou S3 Standard – Accès peu fréquent vers les classes S3 Intelligent-Tiering (Hiérarchisation intelligente), S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent

Lorsque vous effectuez la transition d'objets des classes de stockage S3 standard ou S3 standard – Accès peu fréquent vers les classes S3 Intelligent-Tiering (Hiérarchisation intelligente), S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent, les contraintes de taille d'objet suivantes s'appliquent :

- Objets plus volumineux – Pour les transitions suivantes, la transition d'objets plus volumineux est financièrement plus avantageuse :
  - Des classes de stockage S3 standard ou S3 standard – Accès peu fréquent vers la classe de stockage S3 Intelligent-Tiering (Hiérarchisation intelligente).
  - De la classe de stockage S3 standard vers les classes de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.
- Objets inférieurs à 128 KiB : pour les transitions suivantes, Amazon S3 ne transfère pas les objets dont la taille est inférieure à 128 KiB :
  - Des classes de stockage S3 standard ou S3 standard – Accès peu fréquent vers S3 Intelligent-Tiering ou S3 Glacier Instant Retrieval.
  - De la classe de stockage S3 standard vers les classes de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.

> **Note**
>
> Vous pouvez filtrer les règles de cycle de vie en fonction de la taille des objets.

> **Important**
>
> Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :

- La suppression permanente a priorité sur la transition.
- La transition a priorité sur la création de marqueurs de suppression.
- Si un objet est éligible pour les transitions vers S3 Glacier Flexible Retrieval et S3 standard – Accès peu fréquent (ou S3 unizone – Accès peu fréquent), Amazon S3 choisit la transition vers S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Jours minimum pour la transition vers S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent

Avant d'effectuer la transition d'objets vers les classes de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent, vous devez les stocker au moins 30 jours dans Amazon S3. Par exemple, vous ne pouvez pas créer une règle de cycle de vie pour effectuer la transition d'objets vers la classe de stockage S3 standard – Accès peu fréquent un jour après leur création. Amazon S3 ne prend pas en charge cette transition dans les 30 premiers jours, car les objets les plus récents font souvent l'objet d'un accès plus fréquent ou sont supprimés plus rapidement comparés aux classes de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.

De même, si vous effectuez la transition d'objets anciens (dans des compartiments prenant en charge la gestion des versions), vous ne pouvez effectuer la transition que d'objets anciens d'au moins 30 jours vers une classe de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent. Pour obtenir la liste des durées de stockage minimales pour toutes les classes de stockage, voirComparaison des classes de stockage Amazon S3.

Frais de stockage minimum de 30 jours pour les classes de stockage S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent

Les classes de stockage S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent induisent une facturation d'un temps de stockage de 30 jours minimum. Par conséquent, vous ne pouvez pas spécifier une règle de cycle de vie unique pour une transition S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent et une transition vers une transition S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive lorsque la transition vers S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive survient moins de 30 jours après la transition S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.

Ce délai minimal de 30 jours s'applique également lorsque vous spécifiez une transition depuis la classe de stockage S3 standard – Accès peu fréquent vers la classe de stockage S3 unizone – Accès peu fréquent. Vous pouvez spécifier deux règles pour y parvenir, mais vous payez des frais de stockage minimaux. Pour plus d'informations sur les coûts, consultez Tarification Amazon S3.

Gestion du cycle de vie complet d'un objet

Vous pouvez combiner ces actions de cycle de vie S3 pour gérer le cycle de vie complet d'un objet. Par exemple, admettons que les objets que vous créez ont un cycle de vie bien défini. À la base, les objets sont utilisés fréquemment pendant une période de 30 jours. Ensuite, les objets font l'objet d'accès peu fréquents pendant une période de 90 jours. Par la suite, les objets ne sont plus nécessaires, vous pouvez donc décider de les archiver ou de les supprimer.

Dans ce scénario, vous pouvez créer une règle de cycle de vie S3 vous permettant de spécifier l'action de transition initiale vers la classe de stockage S3 Intelligent-Tiering, S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent, une autre action de transition vers la classe de stockage S3 Glacier Flexible Retrieval à des fins d'archivage et une action d'expiration. Lorsque vous déplacez les objets d'une classe de stockage à une autre, vous économisez sur les coûts de stockage. Pour plus d'informations sur les coûts, consultez Tarification Amazon S3.

Transition vers les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive (archivage d'objets)

En utilisant une configuration S3 Lifecycle, vous pouvez transférer des objets vers les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive à des fins d'archivage. Lorsque vous choisissez les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, vos objets restent dans Amazon S3. Vous ne pouvez pas y accéder directement via le service Amazon S3 Glacier distinct. Pour plus d'informations générales sur S3 Glacier, consultez la section Qu'est-ce qu'Amazon S3 Glacier dans le Guide du développeur Amazon S3 Glacier.

Avant d'archiver des objets, passez en revue les sections suivantes contenant des considérations pertinentes.

Considérations d'ordre général

Les considérations générales suivantes sont à prendre en compte avant d'archiver des objets :

- Les objets chiffrés restent chiffrés tout au long du processus de transition de la classe de stockage.
- Les objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive ne sont pas disponibles en temps réel.

Les objets archivés sont des objets Amazon S3, mais avant de pouvoir accéder à un objet archivé, vous devez d'abord en restaurer une copie temporaire. La copie de l'objet restauré n'est disponible que pendant la durée que vous spécifiez dans la demande de restauration. Après quoi, Amazon S3 supprime la copie temporaire et l'objet reste archivé dans Amazon S3 Glacier Flexible Retrieval.

Vous pouvez restaurer un objet à l'aide de la console Amazon S3 ou par programmation en utilisant les bibliothèques d'encapsulation du AWS SDK ou l'API REST Amazon S3 dans votre code. Pour plus d'informations, consultez Restauration d'un objet archivé.

- Les objets stockés dans la classe de stockage S3 Glacier Flexible Retrieval peuvent uniquement faire l'objet d'une transition vers la classe de stockage S3 Glacier Deep Archive.

  Vous pouvez utiliser une règle de configuration de cycle de vie S3 pour convertir la classe de stockage d'un objet de S3 Glacier Flexible Retrieval vers la classe de stockage S3 Glacier Deep Archive uniquement. Si vous souhaitez modifier la classe de stockage d'un objet stocké dans S3 Glacier Flexible Retrieval en une classe de stockage autre que S3 Glacier Deep Archive, vous devez d'abord utiliser l'opération de restauration pour créer une copie temporaire de l'objet. Utilisez ensuite l'opération de copie pour remplacer l'objet spécifiant S3 standard, S3 Intelligent-Tiering (Hiérarchisation intelligente), S3 standard – Accès peu fréquent, S3 unizone – Accès peu fréquent ou Redondance réduite comme classe de stockage.

- La transition d'objets vers la classe de stockage S3 Glacier Deep Archive est unidirectionnelle.

  Vous ne pouvez pas utiliser une règle de configuration du cycle de vie S3 pour convertir un objet de la classe de stockage S3 Glacier Deep Archive vers toute autre classe de stockage. Si vous souhaitez modifier la classe de stockage d'un objet archivé en une autre classe, vous devez utiliser l'opération de restauration pour effectuer d'abord une copie de l'objet. Utilisez ensuite l'opération de copie pour remplacer l'objet en spécifiant S3 standard, S3 Intelligent-Tiering, S3 standard – Accès peu fréquent, S3 unizone – Accès peu fréquent, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou Stockage à redondance réduite comme classe de stockage.

> **Note**
>
> L'opération de copie des objets restaurés n'est pas prise en charge dans la console Amazon S3 pour les objets des classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. Pour ce type d'opération de copie, utilisez le AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST.

Les objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont visibles et disponibles uniquement via Amazon S3. Ils ne sont pas disponibles via le service Amazon S3 Glacier séparé.

Il s'agit d'objets Amazon S3 et vous pouvez uniquement y accéder à l'aide de la console Amazon S3 ou de l'API Amazon S3. Vous ne pouvez pas accéder aux objets archivés via la console Amazon S3 Glacier séparée ou l'API Amazon S3 Glacier.

Considérations de coût

Si vous envisagez d'archiver des données rarement accédées pendant plusieurs mois ou années, les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive peuvent réduire vos coûts de stockage. Toutefois, pour vous assurer que la classe de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive vous convient, tenez compte des éléments suivants :

- Frais généraux de stockage – Lorsque vous effectuez la transition d'objets vers la classe de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, un volume fixe de stockage est ajouté à chaque objet pour adapter les métadonnées à la gestion de l'objet.
  - Pour chaque objet archivé dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, Amazon S3 utilise 8 Ko de stockage pour le nom de l'objet et d'autres métadonnées. Amazon S3 stocke ces métadonnées pour que vous puissiez obtenir une liste en temps réel de vos objets archivés à l'aide de l'API Amazon S3. Pour plus d'informations, consultez Get Bucket (List Objects). Le tarif S3 Standard vous est facturé pour ce stockage supplémentaire.
  - Pour chaque objet archivé dans S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, Amazon S3 ajoute 32 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive vous est facturé pour ce stockage supplémentaire.

  Si vous archivez de petits objets, tenez compte de ces frais de stockage. Pensez également à regrouper de nombreux petits objets en un plus petit nombre de gros objets afin de réduire les frais généraux.
- Nombre de jours prévu pour la conservation des objets archivés – S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont des solutions d'archivage à long terme. La durée minimale de stockage est de 90 jours pour la classe de stockage S3 Glacier Flexible Retrieval et de 180 jours pour S3 Glacier Deep Archive. La suppression des données archivées dans Amazon S3 Glacier n'entraîne aucuns frais si les objets que vous supprimez sont archivés depuis plus longtemps que

la durée de stockage minimale. Si vous supprimez ou remplacez un objet archivé dans la durée minimale impartie, Amazon S3 facture des frais de suppression anticipés calculés au prorata. Pour plus d'informations sur les frais de suppression anticipée, consultez la question Comment les frais de suppression d'objets datant de moins de 90 jours dans Amazon S3 Glacier sont-ils facturés ? sur le FAQ sur Amazon S3.

- Frais de demande de transition S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive – Chaque objet pour lequel vous effectuez la transition vers la classe de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive constitue une demande de transition. Chaque demande de ce type a un coût. Si vous envisagez de procéder à la transition d'un grand nombre d'objets, tenez compte des coûts de demande. Si vous archivez une combinaison d'objets comprenant de petits objets, en particulier ceux de moins de 128 Ko, nous vous recommandons d'utiliser le filtre de taille des objets du cycle de vie pour exclure les petits objets de votre transition afin de réduire les coûts liés aux demandes. S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive ne bloquent pas automatiquement la transition d'objets de moins de 128 Ko.
- Frais de restauration des données S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive – S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive sont conçus pour un archivage à long terme des données auxquelles vous accédez rarement. Pour plus d'informations sur les frais de restauration des données, consultez la question « Combien coûte l'extraction de données à partir d'Amazon S3 Glacier ? ». sur le FAQ sur Amazon S3. Pour plus d'informations sur la restauration des données depuis Amazon S3 Glacier, consultez Restauration d'un objet archivé.

Lorsque vous archivez des objets dans Amazon S3 Glacier grâce à la gestion de cycle de vie S3, Amazon S3 effectue la transition de ces objets de manière asynchrone. Un délai peut s'écouler entre la date de transition indiquée dans la règle de configuration de cycle de vie S3 et la date de transition physique. Les tarifs Amazon S3 Glacier vous sont facturés selon la date de transition spécifiée dans la règle. Pour plus d'informations, consultez la section Amazon S3 Glacier de la FAQ sur Amazon S3.

La page détaillée du produit Amazon S3 fournit des informations sur la tarification et des exemples de calcul pour l'archivage d'objets Amazon S3. Pour plus d'informations, consultez les rubriques suivantes :

- Comment les frais de stockage pour les objets Amazon S3 archivés dans Amazon S3 Glacier sont-ils calculés ? sur le FAQ sur Amazon S3.
- Comment les frais de suppression d'objets datant de moins de 90 jours dans Amazon S3 Glacier sont-ils calculés ? sur le FAQ sur Amazon S3.
- Combien coûte l'extraction de données à partir d'Amazon S3 Glacier ? sur le FAQ sur Amazon S3.

- Tarification Amazon S3 pour les coûts de stockage concernant différentes classes de stockage.

Restauration d'objets archivés

Les objets archivés ne sont pas accessibles en temps réel. Vous devez d'abord lancer une demande de restauration puis attendre jusqu'à ce qu'une copie temporaire de l'objet soit disponible pour la durée spécifiée dans la demande. Une fois que vous avez reçu une copie temporaire de l'objet restauré, la classe de stockage de l'objet reste S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive. (Une demande d'opération HeadObjectou d'GetObjectAPI renverra S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive comme classe de stockage.)

> **Note**
>
> Lorsque vous restaurez une archive, vous payez à la fois l'archive (tarif S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive) et une copie que vous avez restaurée temporairement (tarif de stockage S3 standard). Pour obtenir des informations sur la tarification, veuillez consulter Tarification Amazon S3.

Vous pouvez restaurer la copie d'un objet par programmation ou à l'aide de la console Amazon S3. Amazon S3 ne traite qu'une seule demande de restauration simultanée par objet. Pour plus d'informations, consultez Restauration d'un objet archivé.

Objets en cours d'expiration

Lorsqu'un objet atteint la fin de sa durée de vie en fonction de la configuration de son cycle de vie, Amazon S3 prend une action en fonction de l'état de gestion des versions S3 dans lequel se trouve le compartiment.

- Compartiment non versionné : Amazon S3 met l'objet en file d'attente pour le supprimer et le supprime de manière asynchrone, le supprimant définitivement.
- Compartiment activé pour la gestion des versions : si la version d'objet actuelle n'est pas un marqueur de suppression, Amazon S3 ajoute un marqueur de suppression avec un ID de version unique. Cela définit la version actuelle comme ancienne et le marqueur de suppression devient la version actuelle.
- Compartiment avec gestion des versions suspendue : Amazon S3 crée un marqueur de suppression avec l'ID de version Null. Ce marqueur de suppression remplace toute version d'objet par un ID de version nul dans la hiérarchie des versions, ce qui supprime l'objet.

Pour un compartiment avec gestion des versions (c'est-à-dire dont la gestion des versions est activée ou suspendue), plusieurs considérations guident la façon dont Amazon S3 gère l'action d'expiration. Pour les compartiments avec gestion des versions activée ou suspendue, les règles suivantes s'appliquent :

- L'expiration d'objet s'applique uniquement à la version actuelle de l'objet (elle n'a aucun impact sur les versions d'objet non actuelles).
- Amazon S3 n'effectue aucune action en présence d'une ou de plusieurs versions d'objet et si le marqueur de suppression est la version actuelle.
- Si la version d'objet actuelle est la seule version d'objet et qu'elle est aussi un marqueur de suppression (également appelé marqueur de suppression d'objet expiré, où toutes les versions d'objet sont supprimées et où il ne reste que le marqueur de suppression), Amazon S3 supprime le marqueur de suppression d'objet expiré. Vous pouvez aussi utiliser l'action d'expiration pour indiquer à Amazon S3 de supprimer tout marqueur de suppression d'objet expiré. Pour obtenir un exemple, consultez Exemple 7 : Suppression des marqueurs de suppression d'objet expiré.
- Vous pouvez utiliser l'élément `NoncurrentVersionExpiration` action pour demander à Amazon S3 de supprimer définitivement les versions non actuelles des objets. Ces objets supprimés ne peuvent pas être récupérés. Vous pouvez baser cette expiration sur un certain nombre de jours écoulés depuis que les objets sont devenus caducs. Outre le nombre de jours, vous pouvez également indiquer un nombre maximum de versions non actuelles à conserver (entre 1 et 100). Cette valeur indique le nombre de nouvelles versions anciennes qui doivent exister pour qu'Amazon S3 puisse effectuer l'action associée sur une version donnée. Pour spécifier le nombre maximum de versions non actuelles, vous devez également fournir un `Filter` élément. Si vous ne spécifiez aucun `Filter` élément, Amazon S3 génère une `InvalidRequest` erreur lorsque vous fournissez un nombre maximum de versions non actuelles. Pour plus d'informations sur l'utilisation de l'élément `NoncurrentVersionExpiration` action, consultezthe section called “Éléments pour décrire les actions du cycle de vie”.

Pour plus d’informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

> **Important**
>
> Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :
>
> - La suppression permanente a priorité sur la transition.

- La transition a priorité sur la création de marqueurs de suppression.
- Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Objets existants et nouveaux

Quand vous ajoutez une configuration de cycle de vie dans un compartiment, les règles de configuration s'appliquent à la fois aux objets existants et à ceux que vous ajouterez ultérieurement. Par exemple, si vous ajoutez une règle de configuration du cycle de vie aujourd'hui avec une action d'expiration qui fait expirer les objets dotés d'un préfixe spécifique 30 jours après leur création, Amazon S3 mettra en file d'attente pour suppression tous les objets existants âgés de plus de 30 jours et portant le préfixe spécifié.

Important

Vous ne pouvez pas utiliser une politique de compartiment pour empêcher les suppressions ou les transitions selon une règle du cycle de vie S3. Par exemple, même si votre politique de compartiment refuse toutes les actions pour tous les principaux, votre configuration S3 Lifecycle fonctionne toujours normalement.

Comment savoir quand les objets vont expirer

Pour connaître la date d'expiration prévue d'un objet, utilisez l'opération HeadObjectou GetObjectAPI. Ces opérations d'API renvoient des en-têtes de réponse qui indiquent la date et l'heure auxquelles l'objet ne peut plus être mis en cache.

Note

- Un certain retard est possible entre la date d'expiration et la date à laquelle Amazon S3 supprime un objet. L'expiration de la durée de stockage associée à un objet ayant expiré n'est pas facturée.

- Avant de mettre à jour, de désactiver ou de supprimer les règles du cycle de vie, utilisez les opérations d'`LIST`API (telles que ListObjectsV2ListObjectVersions, et ListMultipartUploads) ou Inventaire Simple Storage Service (Amazon S3) vérifiez qu'Amazon S3 a transféré et expiré les objets éligibles en fonction de vos cas d'utilisation.

Frais de durée minimale de stockage

Si vous créez une règle d'expiration du cycle de vie S3 entraînant l'expiration des objets appartenant aux classes de stockage S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent depuis moins de 30 jours, vous serez facturé pour la durée de 30 jours. Si vous créez une règle d'expiration de cycle de vie entraînant l'expiration d'objets stockés dans S3 Glacier Flexible Retrieval depuis moins de 90 jours, les 90 jours vous seront facturés. Si vous créez une règle d'expiration du cycle de vie entraînant l'expiration des objets appartenant à la classe de stockage S3 Glacier Deep Archive depuis moins de 180 jours, vous serez facturé pour la durée de 180 jours.

Pour plus d’informations, consultez Tarification Amazon S3.

Configuration du cycle de vie d'un bucket

Cette section explique comment définir une configuration du cycle de vie Amazon S3 sur un compartiment à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), AWS des SDK ou de l'API REST Amazon S3. Pour plus d’informations sur la configuration de cycle de vie S3, consultez Gestion du cycle de vie de votre stockage.

Vous pouvez utiliser des stratégies de cycle de vie pour définir les actions qu'Amazon S3 doit effectuer au cours de la durée de vie d'un objet (par exemple, transférer les objets vers une autre classe de stockage, les archiver ou les supprimer au bout d'une certaine période).

Avant de définir une configuration de cycle de vie, veuillez noter ce qui suit :

Délai de propagation des configurations du cycle de vie

Quand vous ajoutez une configuration de cycle de vie S3 à un compartiment, il y a habituellement un certain délai avant que la nouvelle configuration de cycle de vie ou celle qui a été mise à jour ne soit totalement appliquée à tous les systèmes Amazon S3. Il faut attendre quelques minutes avant que la configuration ne prenne effet. Ce décalage peut également se produire lors de la suppression d'une configuration de cycle de vie S3.

Délai de transition ou d'expiration

Il existe un délai entre le moment où une règle de cycle de vie est satisfaite et le moment où l'action correspondant à la règle est terminée. Supposons, par exemple, qu'un ensemble d'objets soit expiré par une règle de cycle de vie le 1er janvier. Même si la règle d'expiration a été respectée le 1er janvier, Amazon S3 ne supprimera peut-être ces objets que des jours, voire des semaines plus tard. Ce délai est dû au fait que S3 Lifecycle met en file d'attente des objets pour des transitions ou des expirations de manière asynchrone. Toutefois, les modifications apportées à la facturation sont généralement appliquées lorsque la règle du cycle de vie est respectée, même si l'action n'est pas terminée. Pour plus d'informations, consultez la section Modifications apportées à la facturation. Pour surveiller l'effet des mises à jour effectuées par les règles de cycle de vie actives, voir the section called “Comment puis-je contrôler les mesures prises conformément à mes règles de cycle de vie ?”

Désactivation ou suppression des règles de cycle de vie

Lorsque vous désactivez ou supprimez les règles de cycle de vie, Amazon S3 arrête de planifier la suppression ou la transition de nouveaux objets après un bref délai. Tous les objets déjà planifiés sont déprogrammés et ne sont ni supprimés ni transférés.

> **Note**
>
> Avant de mettre à jour, de désactiver ou de supprimer les règles de cycle de vie, utilisez les opérations d'`LIST`API (telles que ListObjectsV2ListObjectVersions, et ListMultipartUploads) ou Inventaire Simple Storage Service (Amazon S3) vérifiez qu'Amazon S3 a transféré et expiré les objets éligibles en fonction de vos cas d'utilisation. Si vous rencontrez des problèmes lors de la mise à jour, de la désactivation ou de la suppression des règles de cycle de vie, consultezRésolution des problèmes de cycle de vie Amazon S3.

Objets existants et nouveaux

Quand vous ajoutez une configuration de cycle de vie dans un compartiment, les règles de configuration s'appliquent à la fois aux objets existants et à ceux que vous ajouterez ultérieurement. Par exemple, si vous ajoutez une règle de configuration du cycle de vie aujourd'hui avec une action d'expiration qui fait expirer les objets dotés d'un préfixe spécifique 30 jours après leur création, Amazon S3 mettra en file d'attente pour suppression tous les objets existants âgés de plus de 30 jours et portant le préfixe spécifié.

Surveillance de l'effet des règles relatives au cycle de vie

Pour surveiller l'effet des mises à jour effectuées par les règles de cycle de vie actives, voir the section called “Comment puis-je contrôler les mesures prises conformément à mes règles de cycle de vie ?”

Modifications apportées à la facturation

Il peut y avoir un décalage entre le moment où les règles de configuration du cycle de vie sont satisfaites et le moment où l'action déclenchée par le respect de la règle est prise. Cependant, les modifications de facturation se produisent dès que la règle de configuration du cycle de vie est satisfaite, même si aucune mesure n'est encore prise.

Par exemple, après le délai d'expiration de l'objet, le stockage ne vous est pas facturé, même si l'objet n'est pas supprimé immédiatement. De même, dès que le délai de transition de l'objet est écoulé, les frais de stockage S3 Glacier Flexible Retrieval vous sont facturés, même si l'objet n'est pas immédiatement transféré vers la classe de stockage S3 Glacier Flexible Retrieval.

Toutefois, les transitions du cycle de vie vers la classe de stockage S3 Intelligent-Tiering constituent une exception. Les modifications de facturation ne se produisent qu'après la transition de l'objet vers la classe de stockage S3 Intelligent-Tiering.

Règles multiples ou contradictoires

Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :

- La suppression permanente a priorité sur la transition.
- La transition a priorité sur la création de marqueurs de suppression.
- Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Utilisation de la console S3

Vous pouvez définir des règles de cycle de vie pour tous les objets ou pour un sous-ensemble d'objets d'un bucket à l'aide d'un préfixe partagé (noms d'objets commençant par une chaîne commune) ou d'une balise. Dans votre règle de cycle de vie, vous pouvez définir des actions

spécifiques aux versions actuelles et non actuelles de l'objet. Pour plus d'informations, consultez les ressources suivantes :

- Gestion du cycle de vie de votre stockage
- Utilisation de la gestion des versions dans les compartiments S3

Pour créer une stratégie de cycle de vie

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez créer une stratégie de cycle de vie.
3. Choisissez l'onglet Management (Gestion), puis choisissez Create lifecycle rule (Créer une règle de cycle de vie).
4. Dans Lifecycle rule name (Nom de la règle du cycle de vie), saisissez un nom pour votre règle.

   Ce nom doit être unique dans le compartiment.
5. Choisissez l'étendue de la règle de cycle de vie :

   - Pour appliquer cette règle de cycle de vie à tous les objets avec un préfixe ou une balise spécifique, choisissez Limiter la portée à des préfixes ou balises spécifiques.
     - Pour limiter l'étendue par préfixe, saisissez le préfixe dans Prefix (Préfixe).
     - Pour limiter l'étendue par balise, choisissez Add tag (Ajouter une balise), puis saisissez la clé et la valeur de la balise.

     Pour en savoir plus sur les préfixes de nom d'objet, veuillez consulter Création de noms de clés d'objet. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.
   - Pour appliquer cette règle de cycle de vie à tous les objets du compartiment, choisissez Cette règle s'applique à tous les objets du compartiment, puis choisissez Je reconnais que cette règle s'applique à tous les objets du compartiment.
6. Pour filtrer une règle par taille d'objet, vous pouvez sélectionner Spécifier la taille minimale de l'objet, Spécifier la taille maximale de l'objet ou les deux options.

- Lorsque vous spécifiez une valeur pour la taille minimale de l'objet ou la taille maximale de l'objet, la valeur doit être supérieure à 0 octet et maximale de 5 To. Vous pouvez spécifier cette valeur en octets, Ko, Mo ou Go.
- Lorsque vous spécifiez les deux valeurs, la taille maximale de l'objet doit être supérieure à la taille minimale de l'objet.

> **Note**
>
> Les filtres Taille d'objet minimale et Taille maximale d'objet excluent les valeurs spécifiées. Par exemple, si vous définissez un filtre pour faire expirer les objets dont la taille minimale est de 128 Ko, les objets dont la taille est exactement de 128 Ko n'expirent pas. Au lieu de cela, la règle s'applique uniquement aux objets dont la taille est supérieure à 128 Ko.

7. Sous Lifecycle rule actions (Actions de règle de cycle de vie), choisissez les actions que votre règle de cycle de vie doit effectuer :

- Transition current versions of objects between storage classes (Transition des versions actuelles des objets entre les classes de stockage)
- Transition previous versions of objects between storage classes (Transition des versions précédentes des objets entre les classes de stockage)
- Expire current versions of objects (Faire expirer les versions actuelles des objets)

> **Note**
>
> Pour les compartiments pour lesquels le versionnage S3 n'est pas activé, l'expiration des versions actuelles entraîne la suppression définitive des objets par Amazon S3. Pour plus d'informations, consultez the section called “Actions du cycle de vie et état du contrôle de version du compartiment”.

- Permanently delete previous versions of objects (Supprimer définitivement les versions précédentes des objets)
- Delete expired delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression expirés ou les chargements partitionnés non terminés)

Selon les actions que vous choisissez, différentes options apparaissent.

8. Pour faire la transition des versions actuelles des objets entre les classes de stockage, sous Transition current versions of objects between storage classes (Transition des versions actuelles des objets entre les classes de stockage):

   a. Dans Transitions de classe de stockage, choisissez la classe de stockage vers laquelle effectuer la transition. Pour obtenir la liste des transitions possibles, voirthe section called “Transitions de cycle de vie prises en charge”. Vous pouvez choisir parmi les classes de stockage suivantes :

      - S3 standard – Accès peu fréquent
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Flexible Retrival
      - S3 Glacier Deep Archive

   b. Dans Days after object creation (Jours après la création de l'objet), entrez le nombre de jours après la création pour la transition de l'objet.

   Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). Vous pouvez définir le transfert des versions actuelles ou précédentes des objets, ou des deux versions à la fois. Le contrôle de version vous permet de conserver plusieurs versions d'un objet au sein d'un même compartiment. Pour plus d’informations sur la gestion des versions, consultez Utilisation de la console S3.

   > **Important**
   >
   > Lorsque vous choisissez les classes de stockage S3 Glacier Flexible Retrieval ou Glacier Deep Archive, vos objets restent dans Amazon S3. Vous ne pouvez pas y accéder directement via le service Amazon S3 Glacier distinct. Pour plus d’informations, consultez Transition des objets à l'aide du cycle de vie Amazon S3.

9. Pour effectuer la transition de versions non actuelles d'objets entre classes de stockage, sous Transition de versions non actuelles d'objets entre classes de stockage :

   a. Dans Transitions de classe de stockage, choisissez la classe de stockage vers laquelle effectuer la transition. Pour obtenir la liste des transitions possibles, voirthe section called “Transitions de cycle de vie prises en charge”. Vous pouvez choisir parmi les classes de stockage suivantes :

- S3 standard – Accès peu fréquent
- S3 Intelligent-Tiering
- S3 One Zone-IA
- S3 Glacier Flexible Retrival
- S3 Glacier Deep Archive

b. Dans Jours après que l'objet ne soit plus actuel, entrez le nombre de jours après sa création pour effectuer la transition de l'objet.

10. Pour faire expirer les versions actuelles des objets, sous Expire current versions of objects (Faire expirer des versions actuelles des objets), dans Number of days after object creation (Nombre de jours après la création de l'objet), entrez le nombre de jours.

> **Important**
>
> Dans un compartiment non versionné, l'action d'expiration entraîne la suppression définitive de l'objet par Amazon S3. Pour en savoir plus sur les actions de cycle de vie, veuillez consulter Éléments pour décrire les actions du cycle de vie.

11. Pour supprimer définitivement les versions précédentes d'objets, sous Permanently delete noncurrent versions of objects (Supprimer définitivement les versions précédentes des objets), dans Days after objects become previous versions (Jours après que les objets deviennent des versions précédentes), entrez le nombre de jours. Vous pouvez éventuellement spécifier le nombre de versions plus récentes à conserver en saisissant une valeur sous Number of newer versions to retain (Nombre de versions plus récentes à conserver).
12. Sous Delete expired delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression expirés ou les chargements en plusieurs parties incomplets), choisissez Delete expired object delete markers (Supprimer les marqueurs de suppression d'objet arrivés à expiration) et Delete incomplete multipart uploads (Supprimer les chargements partitionnés non terminés). Entrez ensuite le nombre de jours après le début du chargement partitionné que vous souhaitez arrêter et nettoyez les chargements en plusieurs parties incomplets.

    Pour en savoir plus sur le chargement partitionné, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.
13. Choisissez Créer une règle.

Si la règle ne contient aucune erreur, Amazon S3 l'active et vous pouvez la voir dans l'onglet Management (Gestion) sous Lifecycle rules (Règles de cycle de vie).

Pour plus d'informations sur les AWS CloudFormation modèles et les exemples, voir Utilisation des AWS CloudFormation modèles et AWS::S3::Bucketdans le Guide de AWS CloudFormation l'utilisateur.

En utilisant le AWS CLI

Vous pouvez utiliser les AWS CLI commandes suivantes pour gérer les configurations du cycle de vie S3 :

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Pour obtenir des instructions sur la configuration du AWS CLI, voirDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

Notez que la configuration de cycle de vie Amazon S3 est un fichier XML. Mais lorsque vous utilisez le AWS CLI, vous ne pouvez pas spécifier le format XML. Vous devez plutôt spécifier le format JSON. Vous trouverez ci-dessous des exemples de configurations de cycle de vie XML et les configurations JSON équivalentes que vous pouvez spécifier dans une AWS CLI commande.

Prenez l'exemple suivant de configuration de cycle de vie S3.

Example Exemple 1

Example

XML

```
<LifecycleConfiguration>
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
           <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
```

```
            <Days>365</Days>
            <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "Filter": {
                "Prefix": "documents/"
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 365,
                    "StorageClass": "GLACIER"
                }
            ],
            "Expiration": {
                "Days": 3650
            },
            "ID": "ExampleRule"
        }
    ]
        }
```

Example Exemple 2

Example

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Rule>
        <ID>id-1</ID>
```

```
        <Expiration>
            <Days>1</Days>
        </Expiration>
        <Filter>
            <And>
                <Prefix>myprefix</Prefix>
                <Tag>
                    <Key>mytagkey1</Key>
                    <Value>mytagvalue1</Value>
                </Tag>
                <Tag>
                    <Key>mytagkey2</Key>
                    <Value>mytagvalue2</Value>
                </Tag>
            </And>
        </Filter>
        <Status>Enabled</Status>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "ID": "id-1",
            "Filter": {
                "And": {
                    "Prefix": "myprefix",
                    "Tags": [
                        {
                            "Value": "mytagvalue1",
                            "Key": "mytagkey1"
                        },
                        {
                            "Value": "mytagvalue2",
                            "Key": "mytagkey2"
                        }
                    ]
                }
            },
            "Status": "Enabled",
```

```
            "Expiration": {
                "Days": 1
            }
        }
    ]
}
```

Vous pouvez tester la commande `put-bucket-lifecycle-configuration` comme suit.

Tester la configuration

1. Enregistrez la configuration du cycle de vie JSON dans un fichier (par exemple, *`lifecycle.json`*).
2. Exécutez la AWS CLI commande suivante pour définir la configuration du cycle de vie de votre bucket. Remplacez *`user input placeholders`* par vos propres informations.

   ```
   $ aws s3api put-bucket-lifecycle-configuration  \
   --bucket DOC-EXAMPLE-BUCKET  \
   --lifecycle-configuration file://lifecycle.json
   ```

3. Pour vérifier, récupérez la configuration du cycle de vie S3 à l'aide de la `get-bucket-lifecycle-configuration` AWS CLI commande suivante :

   ```
   $ aws s3api get-bucket-lifecycle-configuration  \
   --bucket DOC-EXAMPLE-BUCKET
   ```

4. Pour supprimer la configuration du cycle de vie S3, utilisez la `delete-bucket-lifecycle` AWS CLI commande suivante :

   ```
   aws s3api delete-bucket-lifecycle \
   --bucket DOC-EXAMPLE-BUCKET
   ```

Utilisation des AWS SDK

Java

Vous pouvez utiliser le AWS SDK for Java pour gérer la configuration du cycle de vie S3 d'un compartiment. Pour plus d'informations sur la gestion de la configuration du cycle de vie S3, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Quand vous ajoutez une configuration de cycle de vie S3 à un compartiment, Amazon S3 remplace la configuration de cycle de vie actuelle du compartiment, s'il en existe une. Pour mettre à jour une configuration, vous devez la récupérer, effectuer les modifications souhaitées, puis ajouter la configuration révisée dans le compartiment.

L'exemple suivant montre comment utiliser le pour ajouter, mettre AWS SDK for Java à jour et supprimer la configuration du cycle de vie d'un bucket. Cet exemple effectue les opérations suivantes :

- Il ajoute une configuration de cycle de vie à un compartiment.
- Il récupère la configuration de cycle de vie et la met à jour en ajoutant une autre règle.
- Il ajoute la configuration de cycle de vie modifiée au compartiment. Amazon S3 remplace la configuration existante.
- Récupère à nouveau la configuration et vérifie qu'elle contient le bon nombre de règles en imprimant le nombre de règles.
- Il supprime la configuration de cycle de vie et vérifie que celle-ci a été supprimée en tentant de la récupérer à nouveau.

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
```

```
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String bucketName = "*** Bucket name ***";

                // Create a rule to archive objects with the "glacierobjects/"
 prefix to Glacier
                // immediately.
                BucketLifecycleConfiguration.Rule rule1 = new
 BucketLifecycleConfiguration.Rule()
                                .withId("Archive immediately rule")
                                .withFilter(new LifecycleFilter(new
 LifecyclePrefixPredicate("glacierobjects/")))
                                .addTransition(new
 Transition().withDays(0).withStorageClass(StorageClass.Glacier))
                                .withStatus(BucketLifecycleConfiguration.ENABLED);

                // Create a rule to transition objects to the Standard-Infrequent
 Access storage
                // class
                // after 30 days, then to Glacier after 365 days. Amazon S3 will
 delete the
                // objects after 3650 days.
                // The rule applies to all objects with the tag "archive" set to
 "true".
                BucketLifecycleConfiguration.Rule rule2 = new
 BucketLifecycleConfiguration.Rule()
                                .withId("Archive and then delete rule")
                                .withFilter(new LifecycleFilter(new
 LifecycleTagPredicate(new Tag("archive", "true"))))
                                .addTransition(new Transition().withDays(30)

 .withStorageClass(StorageClass.StandardInfrequentAccess))
                                .addTransition(new
 Transition().withDays(365).withStorageClass(StorageClass.Glacier))
                                .withExpirationInDays(3650)
                                .withStatus(BucketLifecycleConfiguration.ENABLED);
```

```
                // Add the rules to a new BucketLifecycleConfiguration.
                BucketLifecycleConfiguration configuration = new
BucketLifecycleConfiguration()
                                .withRules(Arrays.asList(rule1, rule2));

                try {
                        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                                        .withCredentials(new
ProfileCredentialsProvider())
                                        .withRegion(clientRegion)
                                        .build();

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

                        // Retrieve the configuration.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Add a new rule with both a prefix predicate and a tag
predicate.
                        configuration.getRules().add(new
BucketLifecycleConfiguration.Rule().withId("NewRule")
                                        .withFilter(new LifecycleFilter(new
LifecycleAndOperator(
                                                        Arrays.asList(new
LifecyclePrefixPredicate("YearlyDocuments/"),
                                                                        new
LifecycleTagPredicate(new Tag(

   "expire_after",

   "ten_years"))))))
                                        .withExpirationInDays(3650)

.withStatus(BucketLifecycleConfiguration.ENABLED));

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

                        // Retrieve the configuration.
```

```
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration now has three rules.
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);
                        System.out.println("Expected # of rules = 3; found: " +
 configuration.getRules().size());

                        // Delete the configuration.
                        s3Client.deleteBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration has been deleted by
 attempting to retrieve it.
                        configuration =
 s3Client.getBucketLifecycleConfiguration(bucketName);
                        String s = (configuration == null) ? "No configuration
 found." : "Configuration found.";
                        System.out.println(s);
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
 couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
 client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
        }
}
```

.NET

Vous pouvez utiliser le AWS SDK for .NET pour gérer la configuration du cycle de vie S3 sur un bucket. Pour plus d'informations sur la gestion de la configuration du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Quand vous ajoutez une configuration de cycle de vie, Amazon S3 remplace la configuration existante sur le compartiment spécifié. Pour mettre à jour une configuration, vous devez d'abord récupérer la configuration de cycle de vie, effectuer les modifications, puis ajouter la configuration modifiée dans le compartiment.

L'exemple suivant montre comment utiliser le pour ajouter, mettre AWS SDK for .NET à jour et supprimer la configuration du cycle de vie d'un bucket. L'exemple de code effectue les opérations suivantes :

- Il ajoute une configuration de cycle de vie à un compartiment.
- Il récupère la configuration de cycle de vie et la met à jour en ajoutant une autre règle.
- Il ajoute la configuration de cycle de vie modifiée au compartiment. Amazon S3 remplace la configuration de cycle de vie existante.
- Il récupère à nouveau la configuration de cycle de vie et la vérifie en affichant le nombre de règles qu'elle contient.
- Il supprime la configuration de cycle de vie et vérifie qu'elle a été supprimée.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
```

```
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                        {
                            new LifecycleRule
                            {
                                 Id = "Archive immediately rule",
                                 Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                     {
                                         Prefix = "glacierobjects/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Transitions = new List<LifecycleTransition>
                                 {
                                      new LifecycleTransition
                                      {
                                           Days = 0,
                                           StorageClass = S3StorageClass.Glacier
                                      }
                                  },
                            },
                            new LifecycleRule
                            {
                                 Id = "Archive and then delete rule",
                                  Filter = new LifecycleFilter()
                                 {
```

```
                                    LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                    {
                                        Prefix = "projectdocs/"
                                    }
                                },
                                Status = LifecycleRuleStatus.Enabled,
                                Transitions = new List<LifecycleTransition>
                                {
                                     new LifecycleTransition
                                     {
                                          Days = 30,
                                          StorageClass =
S3StorageClass.StandardInfrequentAccess
                                     },
                                     new LifecycleTransition
                                     {
                                       Days = 365,
                                       StorageClass = S3StorageClass.Glacier
                                     }
                                },
                                Expiration = new LifecycleRuleExpiration()
                                {
                                      Days = 3650
                                }
                            }
                        }
                };

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Retrieve an existing configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

                // Add a new rule.
                lifeCycleConfiguration.Rules.Add(new LifecycleRule
                {
                    Id = "NewRule",
                    Filter = new LifecycleFilter()
                    {
                        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                        {
```

```
                            Prefix = "YearlyDocuments/"
                        }
                    },
                    Expiration = new LifecycleRuleExpiration()
                    {
                        Days = 3650
                    }
                });

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Verify that there are now three rules.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
                Console.WriteLine("Expected # of rulest=3; found:{0}",
lifeCycleConfiguration.Rules.Count);

                // Delete the configuration.
                await RemoveLifecycleConfigAsync(client);

                // Retrieve a nonexistent configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
        {

            PutLifecycleConfigurationRequest request = new
PutLifecycleConfigurationRequest
            {
```

```
                BucketName = bucketName,
                Configuration = configuration
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }

        static async Task<LifecycleConfiguration>
 RetrieveLifecycleConfigAsync(IAmazonS3 client)
        {
            GetLifecycleConfigurationRequest request = new
 GetLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }

        static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
        {
            DeleteLifecycleConfigurationRequest request = new
 DeleteLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
    }
}
```

Ruby

Vous pouvez utiliser le AWS SDK for Ruby pour gérer la configuration du cycle de vie S3 sur un compartiment en utilisant la classe AWS::S3::BucketLifecycleConfiguration. Pour plus d'informations sur la gestion de la configuration du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

Utilisation de l'API REST

Les sections suivantes de la Référence des API Amazon Simple Storage Service décrivent l'API REST associée à la configuration de cycle de vie S3.

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Résolution des problèmes liés au cycle de vie S3

Pour les problèmes courants susceptibles de survenir lors de l'utilisation de S3 Lifecycle, consultezthe section called “Résoudre des problèmes de cycle de vie”.

Cycle de vie et autres configurations de compartiment

En plus des configurations de cycle de vie S3, vous pouvez associer d'autres configurations à votre compartiment. Cette section explique comment la configuration de cycle de vie S3 est associée aux autres configurations de compartiment.

Cycle de vie et contrôle de version

Vous pouvez ajouter des configurations de cycle de vie S3 aux compartiments sans gestion des versions et aux compartiments pour lesquels la gestion des versions d'objet est activée. Pour plus d’informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Un compartiment activé pour le contrôle de version maintient une version d'objet actuelle et aucune ou plusieurs versions d'objets anciennes. Vous pouvez définir des règles de cycle de vie séparées pour les versions d'objet actuelles et anciennes.

Pour plus d’informations, consultez Éléments de la configuration du cycle de vie.

> **Important**
>
> Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :
>
> - La suppression permanente a priorité sur la transition.
> - La transition a priorité sur la création de marqueurs de suppression.
> - Si un objet est éligible pour les transitions vers S3 Glacier Flexible Retrieval et S3 standard – Accès peu fréquent (ou S3 unizone – Accès peu fréquent), Amazon S3 choisit la transition vers S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Configuration du cycle de vie sur les compartiments activés pour MFA

La configuration du cycle de vie des compartiments activés pour MFA (authentification multi-facteur) n'est pas prise en charge.

Cycle de vie et journalisation

Les actions du cycle de vie Amazon S3 ne sont pas capturées par la journalisation au niveau de AWS CloudTrail l'objet. CloudTrailcapture les demandes d'API adressées aux points de terminaison Amazon S3 externes, tandis que les actions S3 Lifecycle sont effectuées à l'aide de points de terminaison Amazon S3 internes. Les journaux d'accès au serveur Amazon S3 peuvent être activés dans un compartiment S3 pour enregistrer des actions liées au cycle de vie S3, par exemple la transition d'objet vers une autre classe de stockage et l'expiration d'objet, qui entraînent une suppression permanente ou logique. Pour plus d’informations, consultez the section called “Enregistrement de l'accès au serveur”.

Si la journalisation est activée sur votre compartiment, les journaux d'accès au serveur Amazon S3 signalent les résultats des opérations suivantes.

Journal d'opération	Description
`S3.EXPIRE.OBJECT`	Amazon S3 supprime définitivement l'objet en raison de l'action d'expiration du cycle de vie.
`S3.CREATE.DELETEMARKER`	Amazon S3 supprime logiquement la version actuelle et ajoute un marqueur de suppression dans un compartiment activé pour la gestion des versions.
`S3.TRANSITION_SIA.OBJECT`	Amazon S3 effectue la transition de l'objet vers la classe de stockage S3 standard – Accès peu fréquent.

Journal d'opération	Description
`S3.TRANSITION_ZIA.OBJECT`	Amazon S3 effectue la transition de l'objet vers la classe de stockage S3 unizone – Accès peu fréquent.
`S3.TRANSITION_INT.OBJECT`	Amazon S3 effectue la transition de l'objet vers la classe de stockage S3 Intelligent-Tiering.
`S3.TRANSITION_GIR.OBJECT`	Amazon S3 initie la transition de l'objet vers la classe de stockage S3 Glacier Instant Retrieval .
`S3.TRANSITION.OBJECT`	Amazon S3 initie la transition de l'objet vers la classe de stockage S3 Glacier Flexible Retrieval.
`S3.TRANSITION_GDA.OBJECT`	Amazon S3 initie la transition de l'objet vers la classe de stockage S3 Glacier Deep Archive.
`S3.DELETE.UPLOAD`	Amazon S3 annule un chargement partitionné incomplet.

> **Note**
>
> Les enregistrements des journaux d'accès au serveur Amazon S3 sont généralement fournis sur la base du meilleur effort et ne peuvent pas être utilisés pour une comptabilisation complète de toutes les demandes Amazon S3.

Résolution des problèmes liés au cycle de vie S3

Pour plus d'informations sur la résolution des problèmes courants avec le cycle de vie S3, consultez Résolution des problèmes de cycle de vie Amazon S3.

Plus d'informations

- Éléments de la configuration du cycle de vie

- Transition vers les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive (archivage d'objets)
- Configuration du cycle de vie d'un bucket

Configuration des notifications d'événements de cycle de vie

Vous pouvez configurer une notification d'événement Amazon S3 pour être informé lorsqu'Amazon S3 supprime un objet ou le fait passer à une autre classe de stockage Amazon S3 conformément à une règle de cycle de vie S3.

En utilisant les types d'`LifecycleExpiration`événements, vous pouvez recevoir des notifications chaque fois qu'Amazon S3 supprime un objet en fonction de votre configuration S3 Lifecycle. Le type d'événement `s3:LifecycleExpiration:Delete` vous avertit lorsqu'un objet dans un compartiment non versionné est supprimé. Il vous avertit également lorsqu'une version d'objet est définitivement supprimée par une configuration de cycle de vie S3. Le type d'`s3:LifecycleExpiration:DeleteMarkerCreated`événement vous avertit lorsque S3 Lifecycle crée un marqueur de suppression lorsqu'une version actuelle d'un objet dans un compartiment versionné est supprimée. Pour de plus amples informations, veuillez consulter Suppression des versions d'objet.

En utilisant le type d'`s3:LifecycleTransition`événement, vous pouvez recevoir une notification lorsqu'un objet est transféré d'une classe de stockage Amazon S3 à une autre par une configuration S3 Lifecycle.

Amazon S3 peut publier des notifications d'événements dans une rubrique Amazon Simple Notification Service (Amazon SNS), une file d'attente Amazon Simple Queue Service (Amazon SQS) ou une fonction AWS Lambda . Pour plus d'informations, consultez Notifications d'événements Amazon S3.

Pour obtenir des instructions sur la configuration des notifications d'événements Amazon S3, consultez Activation des notifications d'événements.

Le message suivant est un exemple de message envoyé par Amazon S3 pour publier un événement `s3:LifecycleExpiration:Delete`. Pour en savoir plus, consultez Structure des messages d'événements.

```
{
   "Records":[
      {
```

```
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"LifecycleExpiration:Delete",
         "userIdentity":{
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"example-s3-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::example-s3-bucket"
            },
            "object":{
               "key":"expiration/delete",
               "sequencer":"0055AED6DCD90281E5",
            }
         }
      }
   ]
}
```

Les messages envoyés par Amazon S3 pour publier un `s3:LifecycleTransition` événement incluent également les informations suivantes.

```
"lifecycleEventData":{
    "transitionEventData": {
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Éléments de la configuration du cycle de vie

Rubriques

Vous spécifiez une configuration du cycle de vie Amazon S3 au format XML, composée d'une ou de plusieurs règles de cycle de vie.

```
<LifecycleConfiguration>
    <Rule>
         ...
    </Rule>
    <Rule>
         ...
    </Rule>
</LifecycleConfiguration>
```

Chaque règle se compose des éléments suivants :

- Des métadonnées de règle qui incluent un ID de règle et un statut indiquant si la règle est activée ou désactivée. Si une règle est désactivée, Amazon S3 n'exécute aucune action spécifiée dans la règle.
- Filtre qui identifie les objets auxquels s'applique la règle. Vous pouvez définir un filtre en utilisant la taille de l'objet, le préfixe clé de l'objet, une ou plusieurs balises d'objet ou une combinaison de filtres.
- Une ou plusieurs actions de transition ou d'expiration avec une date ou une durée dans le cycle de vie de l'objet lorsque vous voulez qu'Amazon S3 exécute l'action spécifiée.

Les sections suivantes décrivent les éléments XML dans une configuration de cycle de vie S3. Pour obtenir des exemples de configuration, veuillez consulter Exemples de configuration de cycle de vie S3.

Élément d'ID

Une configuration de cycle de vie S3 peut contenir jusqu'à 1 000 règles. Cette limite n'est pas réglable. L'`<ID>`élément identifie une règle de manière unique. La longueur des ID est limitée à 255 caractères.

Élément de statut

La valeur de l'`<Status>`élément peut être `Enabled` soit`Disabled`. Si une règle est désactivée, Amazon S3 n'exécute aucune action définie dans la règle.

Élément de filtre

Une règle de cycle de vie peut s'appliquer à tous les objets d'un compartiment ou à un sous-ensemble d'entre eux en fonction de l'`<Filter>`élément que vous spécifiez dans la règle de cycle de vie.

Vous pouvez filtrer les objets par préfixe de clé, balises d'objet ou une combinaison des deux (auquel cas, Amazon S3 utilise un opérateur logique `AND` pour combiner les filtres). Considérez les exemples suivants :

- Spécification d'un filtre à l'aide de préfixes clés — Cet exemple montre une règle de cycle de vie S3 qui s'applique à un sous-ensemble d'objets en fonction du préfixe du nom de clé (). `logs/` Par exemple, la règle du cycle de vie s'applique aux objets `logs/mylog.txt`logs/temp1.txt, et`logs/test.txt`. La règle ne s'applique pas à l'objet `example.jpg`.

  ```
  <LifecycleConfiguration>
      <Rule>
          <Filter>
             <Prefix>logs/</Prefix>
          </Filter>
          transition/expiration actions
           ...
      </Rule>
      ...
  </LifecycleConfiguration>
  ```

  Si vous souhaitez appliquer une action de cycle de vie à un sous-ensemble d'objets en fonction de différents préfixes de nom de clé, spécifiez des règles distinctes. Dans chaque règle, spécifiez un filtre basé sur le préfixe. Par exemple, pour décrire une action du cycle de vie des objets dotés des préfixes `projectA/` clés`projectB/`, vous devez spécifier les deux règles suivantes :

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>projectA/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Pour en savoir plus sur les clés d'objet, consultez Création de noms de clés d'objet.

- Spécification d'un filtre basé sur les balises d'objets — Dans l'exemple suivant, la règle Lifecycle spécifie un filtre basé sur une balise (*key*) et une valeur (*value*). La règle s'applique ensuite uniquement à un sous-ensemble d'objets avec la balise spécifique.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
    </Rule>
</LifecycleConfiguration>
```

Vous pouvez spécifier un filtre basé sur plusieurs balises. Vous devez placer les balises dans l'<And>élément, comme indiqué dans l'exemple suivant. La règle indique à Amazon S3 d'exécuter des actions de cycle de vie sur des objets avec deux balises (avec la clé de balise et la valeur spécifiques).

```
<LifecycleConfiguration>
    <Rule>
      <Filter>
         <And>
            <Tag>
               <Key>key1</Key>
               <Value>value1</Value>
            </Tag>
            <Tag>
               <Key>key2</Key>
               <Value>value2</Value>
            </Tag>
             ...
         </And>
      </Filter>
      transition/expiration actions
    </Rule>
</Lifecycle>
```

La règle de cycle de vie s'applique aux objets qui ont tous deux des balises spécifiées. Amazon S3 effectue une opération logique AND. Notez ce qui suit :

- Chaque balise doit correspondre exactement à la fois à la clé et à la valeur. Si vous spécifiez uniquement un <Key> élément et aucun <Value> élément, la règle ne s'applique qu'aux objets correspondant à la clé de balise et pour lesquels aucune valeur n'est spécifiée.
- La règle s'applique à un sous-ensemble d'objets dont toutes les étiquettes sont spécifiées dans la règle. Si un objet dispose d'étiquettes supplémentaires spécifiées, la règle continue à s'appliquer.

> **Note**
>
> Lorsque vous spécifiez plusieurs balises dans un filtre, chaque clé de balise doit être unique.

- Spécification d'un filtre basé à la fois sur le préfixe et sur une ou plusieurs balises : dans une règle de cycle de vie, vous pouvez spécifier un filtre basé à la fois sur le préfixe clé et sur une ou plusieurs balises. Encore une fois, vous devez envelopper tous ces éléments filtrants dans l'<And>élément, comme suit :

```
<LifecycleConfiguration>
```

```
    <Rule>
        <Filter>
          <And>
             <Prefix>key-prefix</Prefix>
             <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
             </Tag>
              ...
          </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Amazon S3 combine ces filtres en utilisant une logiqueAND. En d'autres termes, la règle s'applique au sous-ensemble d'objets dotés du préfixe de clé spécifié et des balises spécifiées. Un filtre peut avoir seulement un préfixe et zéro, une ou plusieurs balises.

- Vous pouvez spécifier un filtre vide, auquel cas la règle s'applique à tous les objets dans le compartiment.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- Pour filtrer une règle par taille d'objet, vous pouvez spécifier une taille minimale (`ObjectSizeGreaterThan`) ou une taille maximale (`ObjectSizeLessThan`), ou vous pouvez spécifier une plage de tailles d'objet.

Les valeurs de taille d'objet sont exprimées en octets. La taille maximale du filtre est de 5 To. Certaines classes de stockage sont soumises à des limites de taille d'objet minimale. Pour plus d'informations, consultez Comparaison des classes de stockage Amazon S3.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> Les `ObjectSizeLessThan` filtres `ObjectSizeGreaterThan` et excluent les valeurs spécifiées. Par exemple, si vous définissez des objets d'une taille de 128 Ko à 1024 Ko pour passer de la classe de stockage S3 Standard à la classe de stockage S3 Standard-IA, les objets d'une taille exacte de 1024 Ko et 128 Ko ne passeront pas à S3 Standard-IA. Au lieu de cela, la règle s'appliquera uniquement aux objets dont la taille est supérieure à 128 Ko et inférieure à 1024 Ko.

Si vous spécifiez une plage de tailles d'objet, l'entier `ObjectSizeGreaterThan` doit être inférieur à la valeur `ObjectSizeLessThan`. Si vous utilisez plusieurs filtres, vous devez les envelopper dans un élément <And>. L'exemple suivant montre comment spécifier des objets dans une plage comprise entre 500 octets et 64 000 octets.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
```

```
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Éléments pour décrire les actions du cycle de vie

Vous pouvez indiquer à Amazon S3 d'exécuter des actions spécifiques dans le cycle de vie d'un objet en spécifiant une ou plusieurs des actions prédéfinies dans une règle de cycle de vie S3. L'effet de ces actions dépend de l'état du contrôle de version de votre compartiment.

- **`Transition`élément d'action** : vous spécifiez l'`Transition`action permettant de faire passer les objets d'une classe de stockage à une autre. Pour en savoir plus sur la transition d'objets, consultez Transitions prises en charge et contraintes connexes. Lorsqu'une date ou une durée spécifiée dans le cycle de vie d'un objet est atteinte, Amazon S3 effectue la transition.

  Pour un compartiment activé pour le contrôle de version (compartiment activé pour le contrôle de version ou suspendu pour le contrôle de version), l'action `Transition` s'applique à la version d'objet actuelle. Pour gérer des versions anciennes, Amazon S3 définit l'action `NoncurrentVersionTransition` (décrite ultérieurement dans cette rubrique).

- **`Expiration`élément d'action** : l'`Expiration`action fait expirer les objets identifiés dans la règle et s'applique aux objets éligibles dans toutes les classes de stockage Amazon S3. Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). Amazon S3 rend tous les objets expirés indisponibles. Que les objets soient supprimés définitivement ou pas dépend de l'état de contrôle de version du compartiment.
  - Compartiment non versionné : l'`Expiration`action entraîne la suppression définitive de l'objet par Amazon S3.
  - Compartiment activé pour la gestion des versions – Pour un compartiment activé pour la gestion des versions (c'est-à-dire dont la gestion des versions est activée ou désactivée), plusieurs considérations guident la façon dont Amazon S3 gère l'action `Expiration`. Pour les compartiments avec gestion des versions activée ou suspendue, les règles suivantes s'appliquent :
    - L'action `Expiration` s'applique uniquement à la version actuelle (elle n'a aucun impact sur les versions d'objet anciennes).

- Amazon S3 n'effectue aucune action en présence d'une ou de plusieurs versions d'objet et si le marqueur de suppression est la version actuelle.
- Si la version d'objet actuelle est la seule version d'objet et qu'elle est aussi un marqueur de suppression (également appelé marqueur de suppression d'objet expiré, où toutes les versions d'objet sont supprimées et où il ne reste que le marqueur de suppression), Amazon S3 supprime le marqueur de suppression d'objet expiré. Vous pouvez aussi utiliser l'action d'expiration pour indiquer à Amazon S3 de supprimer tout marqueur de suppression d'objet expiré. Pour voir un exemple, consultez Exemple 7 : Suppression des marqueurs de suppression d'objet expiré.

Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

Tenez également compte de ce qui suit lors de la configuration d'Amazon S3 pour gérer l'expiration :

- Compartiment activé pour le contrôle de version

  Si la version d'objet actuelle n'est pas un marqueur de suppression, Amazon S3 ajoute un marqueur de suppression avec un ID de version unique. Cela définit la version actuelle comme ancienne et le marqueur de suppression devient la version actuelle.

- Compartiment suspendu pour le contrôle de version

  Dans un compartiment suspendu à la gestion des versions, l'action d'expiration oblige Amazon S3 à créer un marqueur de suppression avec `null` comme ID de version. Ce marqueur de suppression remplace toute version d'objet par un ID de version nul dans la hiérarchie des versions, ce qui supprime l'objet.

De plus, Amazon S3 fournit les actions suivantes que vous pouvez utiliser pour gérer des versions d'objet anciennes dans un compartiment activé pour la gestion des versions (c'est-à-dire, un compartiment dont la gestion des versions est activée ou désactivée).

- **`NoncurrentVersionTransition`**élément d'action — Utilisez cette action pour spécifier à quel moment Amazon S3 fait passer les objets à la classe de stockage spécifiée. Vous pouvez baser cette expiration sur un certain nombre de jours écoulés depuis que les objets sont devenus caducs. Outre le nombre de jours, vous pouvez également indiquer un nombre maximum de versions non actuelles à conserver (entre 1 et 100). Cette valeur détermine le nombre de nouvelles versions non actuelles qui doivent exister avant qu'Amazon S3 puisse effectuer l'action associée sur une

version donnée. Amazon S3 transférera toutes les versions non actuelles supplémentaires au-delà du nombre spécifié à conserver.

Pour spécifier le nombre maximum de versions non actuelles, vous devez également fournir un `Filter` élément. Si vous ne spécifiez aucun `Filter` élément, Amazon S3 génère une `InvalidRequest` erreur lorsque vous fournissez un nombre maximum de versions non actuelles.

Pour en savoir plus sur la transition d'objets, consultez Transitions prises en charge et contraintes connexes. Pour plus de détails sur la façon dont Amazon S3 calcule la date lorsque vous spécifiez le nombre de jours dans l'action `NoncurrentVersionTransition`, consultez Règles de cycle de vie : en fonction de l'âge de l'objet.

- **`NoncurrentVersionExpiration`**élément d'action — Utilisez cette action pour demander à Amazon S3 de supprimer définitivement les versions non actuelles des objets. Ces objets supprimés ne peuvent pas être récupérés. Vous pouvez baser cette expiration sur un certain nombre de jours écoulés depuis que les objets sont devenus caducs. Outre le nombre de jours, vous pouvez également indiquer un nombre maximum de versions non actuelles à conserver (entre 1 et 100). Cette valeur indique le nombre de nouvelles versions anciennes qui doivent exister pour qu'Amazon S3 puisse effectuer l'action associée sur une version donnée. Amazon S3 supprimera définitivement toutes les versions non actuelles supplémentaires au-delà du nombre spécifié à conserver.

  Pour spécifier le nombre maximum de versions non actuelles, vous devez également fournir un `Filter` élément. Si vous ne spécifiez aucun `Filter` élément, Amazon S3 génère une `InvalidRequest` erreur lorsque vous fournissez un nombre maximum de versions non actuelles.

  La suppression retardée d'objets anciens peut être utile lorsque vous avez besoin de corriger des suppressions ou des remplacements accidentels. Par exemple, vous pouvez configurer une règle d'expiration pour supprimer les versions anciennes cinq jours après qu'elles soient devenues anciennes. Supposons, par exemple, que le 1er janvier 2014 à 10 h 30 UTC, vous créiez un objet appelé `photo.gif` (ID de version 111111). Le 02/01/2014 à 11 h 30 UTC, vous supprimez accidentellement `photo.gif` (ID de version 111111), ce qui crée un marqueur de suppression avec un nouvel ID de version (tel que l'ID de version 4857693). Vous avez à présent cinq jours pour récupérer la version d'origine de `photo.gif` (ID de version 111111) avant que la suppression ne soit définitive. Le 08/01/2014 à 00:00 UTC, la règle du cycle de vie pour l'expiration s'exécute et est définitivement supprimée `photo.gif` (ID de version 111111), cinq jours après qu'elle soit devenue une version obsolète.

Pour plus de détails sur la façon dont Amazon S3 calcule la date lorsque vous spécifiez le nombre de jours dans une action `NoncurrentVersionExpiration`, consultez Règles de cycle de vie : en fonction de l'âge de l'objet.

> **Note**
>
> Les configurations du cycle de vie d'expiration des objets ne suppriment pas les téléchargements partitionnés incomplets. Pour supprimer les téléchargements partitionnés incomplets, vous devez utiliser l'action de configuration `AbortIncompleteMultipartUpload` du cycle de vie décrite plus loin dans cette section.

Outre les actions de transition et d'expiration, vous pouvez utiliser les actions de configuration du cycle de vie suivantes pour demander à Amazon S3 d'arrêter les téléchargements partitionnés incomplets ou de supprimer les marqueurs de suppression d'objets expirés :

- **`AbortIncompleteMultipartUpload`**élément d'action : utilisez cet élément pour définir la durée maximale (en jours) pendant laquelle vous souhaitez autoriser les téléchargements partitionnés en cours. Si les téléchargements partitionnés applicables (déterminés par le nom de clé `prefix` spécifié dans la règle du cycle de vie) ne sont pas correctement terminés dans le délai prédéfini, Amazon S3 arrête les téléchargements partitionnés incomplets. Pour plus d'informations, consultez Interruption d'un chargement partitionné.

  > **Note**
  >
  > Vous ne pouvez pas spécifier cette action du cycle de vie dans une règle dotée d'un filtre utilisant des balises d'objet.

- **`ExpiredObjectDeleteMarker`**élément d'action — Dans un compartiment activé pour la gestion des versions, un marqueur de suppression ne contenant aucune version non courante est appelé marqueur de suppression d'objet expiré. Vous pouvez utiliser cette action du cycle de vie pour demander à Amazon S3 de supprimer les marqueurs de suppression d'objets expirés. Pour obtenir un exemple, consultez Exemple 7 : Suppression des marqueurs de suppression d'objet expiré.

> **Note**
>
> Vous ne pouvez pas spécifier cette action du cycle de vie dans une règle dotée d'un filtre utilisant des balises d'objet.

Comment Amazon S3 calcule le temps depuis lequel un objet est ancien

Un compartiment sur lequel la gestion des versions est activée peut contenir de nombreuses versions d'un objet. Il y a toujours une version actuelle et éventuellement une ou plusieurs versions anciennes. Chaque fois que vous chargez un objet, la version actuelle est retenue comme version ancienne et la version nouvellement ajoutée, le successeur, devient la version actuelle. Pour déterminer le nombre de jours depuis lequel un objet est ancien, Amazon S3 regarde la date de création de son successeur. Amazon S3 utilise le nombre de jours depuis la création de son successeur comme le nombre de jours depuis lequel un objet est ancien.

> **Restauration des versions précédentes d'un objet lorsque les configurations de cycle de vie S3 sont utilisées**
>
> Comme expliqué dansRestauration des versions précédentes, vous pouvez utiliser l'une des deux méthodes suivantes pour récupérer les versions précédentes d'un objet :
>
> - Méthode 1 — Copiez une version non actuelle de l'objet dans le même compartiment. L'objet copié devient la version actuelle de cet objet et toutes les versions d'objet sont préservées.
> - Méthode 2 — Supprime définitivement la version actuelle de l'objet. Lorsque vous supprimez la version d'objet actuelle, vous transformez la version ancienne en version actuelle de cet objet.
>
> Lorsque vous utilisez les règles de configuration du cycle de vie S3 avec des compartiments compatibles avec la gestion des versions, nous vous recommandons, comme bonne pratique, d'utiliser la méthode 1.
> Le cycle de vie S3 fonctionne sous un modèle éventuellement cohérent. Une version actuelle que vous avez définitivement supprimée peut ne pas disparaître tant que les modifications ne se sont pas propagées à tous les systèmes Amazon S3. (Il se peut donc qu'Amazon S3 ne soit temporairement pas au courant de cette suppression.) Entre temps, la règle de cycle de vie que vous avez configurée pour l'expiration des objets anciens peut supprimer

définitivement des objets anciens, y compris celui que vous souhaitez restaurer. Copier l'ancienne version, comme recommandé dans la méthode 1, est donc l'alternative la plus sûre.

Actions du cycle de vie et état du contrôle de version du compartiment

Règles de cycle de vie : en fonction de l'âge de l'objet

Vous pouvez spécifier une période, exprimée en jours à compter de la création (ou de la modification) de l'objet, pendant laquelle Amazon S3 peut effectuer l'action spécifiée.

Lorsque vous spécifiez le nombre de jours dans les actions `Transition` et `Expiration` dans une configuration de cycle de vie S3, notez ce qui suit :

- La valeur que vous spécifiez est le nombre de jours écoulés depuis la création de l'objet pendant lesquels l'action aura lieu.
- Amazon S3 calcule l'heure en ajoutant le nombre de jours spécifié dans la règle à l'heure de création de l'objet et en arrondissant l'heure obtenue au jour suivant à minuit UTC. Par exemple, si un objet a été créé le 15/01/2014 à 10h30 UTC et que vous spécifiez 3 jours dans une règle de transition, la date de transition de l'objet sera calculée comme suit : 19/01/2014 00:00 UTC.

> **Note**
>
> Amazon S3 conserve uniquement la dernière date de modification pour chaque objet. Par exemple, la console Amazon S3 affiche la date de dernière modification dans le volet Propriétés de l'objet. Lorsque vous créez un nouvel objet pour la première fois, cette date correspond à la date de création de l'objet. Si vous remplacez l'objet, la date change en conséquence. Par conséquent, la date de création est synonyme de la date de dernière modification.

Lorsqu'un nombre de jours est spécifié dans les actions `NoncurrentVersionTransition` et `NoncurrentVersionExpiration` dans une configuration du cycle de vie, notez ce qui suit :

- La valeur que vous spécifiez est le nombre de jours à compter du moment où la version de l'objet devient caduque (c'est-à-dire lorsque l'objet est remplacé ou supprimé) pendant lesquels Amazon S3 exécute l'action sur l'objet ou les objets spécifiés.

- Amazon S3 calcule l'heure en ajoutant le nombre de jours spécifié dans la règle à l'heure à laquelle la nouvelle version suivante de l'objet est créée et en arrondissant le délai obtenu au jour suivant à minuit UTC. Par exemple, dans votre bucket, supposons que vous disposiez de la version actuelle d'un objet créé le 1er janvier 2014 à 10 h 30 UTC. Si la nouvelle version de l'objet qui remplace la version actuelle est créée le 15/01/2014 à 10h30 UTC, et que vous spécifiez 3 jours dans une règle de transition, la date de transition de l'objet est calculée comme suit : 19/01/2014 00:00 UTC.

Règles de cycle de vie : en fonction d'une date spécifique

Lorsqu'une action est spécifiée dans une règle de cycle de vie S3, vous pouvez spécifier la date à laquelle vous souhaitez qu'Amazon S3 exécute l'action. Lorsque la date spécifiée arrive, Amazon S3 applique l'action à tous les objets qualifiés (en fonction des critères de filtre).

Si vous spécifiez une action S3 Lifecycle avec une date antérieure, tous les objets qualifiés deviennent immédiatement éligibles à cette action du cycle de vie.

> **Important**
>
> L'action basée sur la date n'est pas une action unique. Amazon S3 continue à appliquer l'action basée sur la date même après que la date est passée, tant que l'état de la règle est `Enabled`.
>
> Supposons, par exemple, que vous spécifiez une `Expiration` action basée sur la date pour supprimer tous les objets (supposons qu'aucun filtre n'est spécifié dans la règle). À la date spécifiée, Amazon S3 fait expirer tous les objets du compartiment. Amazon S3 continue également d'expirer tous les nouveaux objets que vous créez dans le compartiment. Pour arrêter l'action du cycle de vie, vous devez soit supprimer l'action de la règle du cycle de vie, soit désactiver la règle, soit supprimer la règle de la configuration du cycle de vie.

La valeur de date doit être conforme au format ISO 8601. L'heure indique toujours minuit UTC.

> **Note**
>
> Vous ne pouvez pas créer de règles de cycle de vie basées sur des dates à l'aide de la console Amazon S3, mais vous pouvez afficher, désactiver ou supprimer de telles règles.

Exemples de configuration de cycle de vie S3

Cette section fournit des exemples de configuration de cycle de vie S3. Chaque exemple indique comment vous pouvez spécifier XML dans chaque exemple de scénarios.

Rubriques

Exemple 1 : Spécification d'un filtre

Chaque règle de cycle de vie S3 comprend un filtre que vous pouvez utiliser pour identifier un sous-ensemble d'objets dans votre compartiment auquel s'applique la règle de cycle de vie S3. Les configurations S3 de cycle de vie suivantes présentent des exemples de spécification d'un filtre.

- Dans cette règle de configuration de cycle de vie S3, le filtre spécifie un préfixe de clé (`tax/`). Par conséquent, la règle s'applique aux objets avec le préfixe de nom de clé `tax/`, comme `tax/doc1.txt` et `tax/doc2.txt`.

  La règle spécifie deux actions qui demandent à Amazon S3 de procéder comme suit :

  - La transition d'objets vers la classe de stockage S3 Glacier Flexible Retrieval 365 jours (un an) après leur création.
  - La suppression d'objets (l'action `Expiration`) 3 650 jours (10 ans) après leur création.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
```

```
    <Filter>
       <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Au lieu de spécifier l'âge de l'objet en termes de jours après sa création, vous pouvez spécifier une date pour chaque action. Cependant, vous ne pouvez pas utiliser à la fois `Date` et `Days` dans la même règle.

- Si vous souhaitez que la règle de cycle de vie S3 s'applique à tous les objets dans le compartiment, spécifiez un préfixe vide. Dans la configuration suivante, la règle spécifie une action `Transition` qui indique à Amazon S3 d'effectuer la transition des objets vers la classe de stockage S3 Glacier Flexible Retrieval 0 jour après leur création. Cette règle signifie que les objets peuvent être archivés dans S3 Glacier Flexible Retrieval à minuit UTC après leur création. Pour d'informations sur les contraintes de cycle de vie, consultez Constraints.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Vous pouvez spécifier un préfixe de nom de clé, ou aucun, et plusieurs balises d'objets dans un filtre, ou aucune. L'exemple de code suivant applique la règle de cycle de vie S3 à un sous-ensemble d'objets avec le préfixe de clé `tax/` et aux objets ayant deux balises avec une clé et

une valeur spécifiques. Si vous spécifiez plusieurs filtres, vous devez inclure l'élément <And> comme illustré (Amazon S3 applique un opérateur logique AND pour combiner les conditions de filtre spécifiées).

```
...
<Filter>
   <And>
      <Prefix>tax/</Prefix>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

- Vous pouvez filtrer des objets en fonction uniquement des balises. Par exemple, la règle de cycle de vie S3 suivante s'applique aux objets ayant les deux balises spécifiées (elle ne spécifie pas de préfixe).

```
...
<Filter>
   <And>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

> **Important**
>
> Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :
>
> - La suppression permanente a priorité sur la transition.
> - La transition a priorité sur la création de marqueurs de suppression.
> - Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.
>
> Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Exemple 2 : Désactivation d'une règle de cycle de vie

Vous pouvez désactiver temporairement une règle du cycle de vie S3. La configuration de cycle de vie S3 suivante spécifie deux règles :

- La règle 1 indique à Amazon S3 d'effectuer la transition des objets avec le préfixe `logs/` vers la classe de stockage S3 Glacier Flexible Retrieval peu de temps après leur création.
- La règle 2 indique à Amazon S3 d'effectuer la transition des objets avec le préfixe `documents/` vers la classe de stockage S3 Glacier Flexible Retrieval peu de temps après leur création.

Dans la configuration, la règle 1 est activée et la règle 2 est désactivée. Amazon S3 ignore les règles désactivées.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
```

```
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule2</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Disabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Exemple 3 : hiérarchisation de la classe de stockage sur la durée de vie d'un objet

Dans cet exemple, vous utilisez une configuration de cycle de vie S3 pour faire passer des objets à la classe de stockage inférieure pendant leur cycle de vie. Ce choix peut permettre de réduire les coûts de stockage. Pour plus d'informations sur la tarification, consultez Tarification Amazon S3.

L'exemple suivant de configuration de cycle de vie S3 spécifie une règle qui s'applique aux objets avec le préfixe de nom de clé `logs/`. La règle spécifie les deux actions suivantes :

- Deux actions de transition :
  - La transition d'objets vers la classe de stockage S3 standard – Accès peu fréquent 30 jours après leur création.
  - La transition d'objets vers la classe de stockage S3 Glacier Flexible Retrieval 90 jours après leur création.
- Une action d'expiration qui indique à Amazon S3 de supprimer les objets un an après leur création.

```
<LifecycleConfiguration>
  <Rule>
    <ID>example-id</ID>
    <Filter>
       <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
```

```
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Note

Vous pouvez utiliser une règle pour décrire toutes les actions de cycle de vie S3 si toutes les actions s'appliquent au même ensemble d'objets (identifié par le filtre). Sinon, vous pouvez ajouter plusieurs règles, chacune spécifiant un filtre différent.

Important

Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :

- La suppression permanente a priorité sur la transition.
- La transition a priorité sur la création de marqueurs de suppression.
- Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Exemple 4 : Spécification de plusieurs règles

Vous pouvez spécifier plusieurs règles si vous souhaitez effectuer différentes actions de cycle de vie S3 pour différents d'objets. La configuration de cycle de vie S3 suivante a deux règles :

- La règle 1 s'applique aux objets ayant le préfixe de nom de clé `classA/`. Elle indique à Amazon S3 d'effectuer la transition d'objets vers la classe de stockage S3 Glacier Flexible Retrieval un an après leur création et de faire expirer ces objets 10 ans après leur création.
- La règle 2 s'applique aux objets ayant le préfixe de nom de clé `classB/`. Elle indique à Amazon S3 d'effectuer la transition d'objets vers la classe de stockage S3 standard – Accès peu fréquent 90 jours après leur création et de les supprimer un an après leur création.

```
<LifecycleConfiguration>
    <Rule>
        <ID>ClassADocRule</ID>
        <Filter>
           <Prefix>classA/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
    <Rule>
        <ID>ClassBDocRule</ID>
        <Filter>
            <Prefix>classB/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <Expiration>
             <Days>365</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Important

Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :

- La suppression permanente a priorité sur la transition.
- La transition a priorité sur la création de marqueurs de suppression.
- Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.

Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3

Vous pouvez définir une configuration de cycle de vie S3 dans laquelle vous spécifiez des préfixes ou des actions se chevauchant.

Généralement, le cycle de vie S3 optimise le coût. Par exemple, si deux stratégies d'expiration se chevauchent, la stratégie d'expiration la plus courte est appliquée pour que des données ne soient pas stockées plus longtemps que prévu. De même, si deux stratégies de transition se chevauchent, le cycle de vie S3 effectue la transition de vos objets vers la classe de stockage au coût le plus bas.

Dans les deux cas, le cycle de vie S3 tente de choisir ce qui est le moins cher pour vous. La classe de stockage S3 Intelligent-Tiering (Hiérarchisation intelligente) constitue une exception à cette règle générale. Le cycle de vie S3 privilégie la classe de stockage S3 Intelligent-Tiering à toutes les autres classes de stockage, à l'exception des classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.

Les exemples suivants illustrent comment Amazon S3 résout les conflits potentiels.

Example 1 : Chevauchement de préfixes (aucun conflit)

L'exemple de configuration suivant comporte deux règles qui indiquent des préfixes qui se chevauchent comme suit :

- La première règle spécifie un filtre vide, indiquant tous les objets du compartiment.
- La deuxième règle spécifie un préfixe de nom de clé (`logs/`), indiquant uniquement un sous-ensemble d'objets.

La règle 1 demande à Amazon S3 de supprimer tous les objets un an après leur création. La règle 2 demande à Amazon S3 d'effectuer la transition d'un sous-ensemble d'objets vers la classe de stockage S3 standard – Accès peu fréquent 30 jours après la création.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>30</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Comme il n'y a pas de conflit dans ce cas, Amazon S3 effectue la transition des objets avec le préfixe `logs/` vers la classe de stockage S3 standard – Accès peu fréquent 30 jours après leur création. Les objets sont ensuite supprimés un an après leur création.

Example 2 : Actions de cycle de vie contradictoires

Dans cet exemple de configuration, il existe deux règles qui indiquent à Amazon S3 d'effectuer en même temps deux actions différentes sur le même ensemble d'objets pendant la durée de vie des objets :

- Les deux règles spécifient le même préfixe de nom de clé et les deux règles s'appliquent par conséquent au même ensemble d'objets.
- Les deux règles spécifient les mêmes 365 jours après la création de l'objet lorsque les règles s'appliquent.
- Une règle indique à Amazon S3 d'effectuer la transition d'objets vers la classe de stockage S3 standard – Accès peu fréquent et une autre règle veut qu'Amazon S3 fasse expirer les objets en même temps.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>365</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Dans ce cas, étant donné que vous voulez que les objets expirent (soient supprimés), il est inutile de modifier la classe de stockage. Par conséquent, Amazon S3 choisit l'action d'expiration sur ces objets.

Example 3 : Chevauchement de préfixes entraînant des actions de cycle de vie contradictoires

Dans cet exemple, la configuration comporte deux règles qui indiquent des préfixes qui se chevauchent comme suit :

- La règle 1 spécifie un préfixe vide (indiquant tous les objets).
- La règle 2 spécifie un préfixe de nom de clé (`logs/`) qui identifie un sous-ensemble de tous les objets.

Pour le sous-ensemble d'objets avec le préfixe de nom de clé `logs/`, les actions de cycle de vie S3 dans les deux règles s'appliquent. Une règle indique à Amazon S3 d'effectuer la transition des objets 10 jours après leur création et une autre règle indique à Amazon S3 d'effectuer la transition des objets 365 jours après leur création.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>10</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>365</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Dans ce cas, Amazon S3 choisit d'effectuer leur transition 10 jours après leur création.

Example 4 : Filtrage basé sur des balises et actions du cycle de vie contradictoires générées

Supposons que vous avez la configuration de cycle de vie S3 suivante qui a deux règles, chacune spécifiant un filtre de balise :

- La règle 1 spécifie un filtre basé sur une balise (`tag1/value1`). Cette règle indique à Amazon S3 d'effectuer la transition des objets vers la classe de stockage S3 Glacier Flexible Retrieval 365 jours après leur création.
- La règle 2 spécifie un filtre basé sur une balise (`tag2/value2`). Cette règle indique à Amazon S3 de faire expirer les objets 14 jours après leur création.

La configuration de cycle de vie S3 est illustrée dans l'exemple suivant.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Tag>
         <Key>tag1</Key>
         <Value>value1</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>GLACIER<StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Tag>
         <Key>tag2</Key>
         <Value>value2</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>14</Days>
    </Expiration>
   </Rule>
</LifecycleConfiguration>
```

Si un objet possède les deux balises, Amazon S3 doit décider de la règle à suivre. Dans ce cas, Amazon S3 fait expirer l'objet 14 jours après sa création. L'objet est supprimé et, par conséquent, l'action de transition ne s'applique pas.

> **Important**
>
> Lorsqu'une configuration S3 Lifecycle comporte plusieurs règles, un objet peut devenir éligible à plusieurs actions du cycle de vie S3 le même jour. Dans de tels cas, Amazon S3 suit les règles générales suivantes :
>
> - La suppression permanente a priorité sur la transition.
> - La transition a priorité sur la création de marqueurs de suppression.
> - Lorsqu'un objet est éligible à la fois à une transition S3 Glacier Flexible Retrieval et à une transition S3 Standard-IA (ou S3 One Zone-IA), Amazon S3 choisit la transition S3 Glacier Flexible Retrieval.
>
> Pour obtenir des exemples, consultez Exemple 5 : Chevauchement de filtres, actions de cycle de vie contradictoires et gestion des compartiments non versionnés par Amazon S3.

Exemple 6 : Spécification d'une règle de cycle de vie pour un compartiment activé pour la gestion des versions

Imaginez que vous avez un compartiment sur lequel la gestion des versions est activée. Dans ce cas, vous avez pour chaque objet une version actuelle et éventuellement une ou plusieurs versions anciennes. (Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3.) Dans cet exemple, vous souhaitez conserver un an d'historique et supprimer les versions anciennes. Les configurations S3 Lifecycle permettent de conserver de 1 à 100 versions de n'importe quel objet.

Afin de réaliser des économiser sur les coûts de stockage, vous souhaitez déplacer les versions anciennes vers S3 Glacier Flexible Retrieval 30 jours après qu'elles sont devenues anciennes (en supposant que ces objets anciens sont des données froides auxquelles vous n'avez pas besoin d'accéder en temps réel). En outre, vous vous attendez à ce que la fréquence d'accès aux versions actuelles diminue 90 jours après leur création. Vous pouvez donc choisir de déplacer ces objets vers la classe de stockage S3 Standard-IA.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
```

```
        <Filter>
           <Prefix></Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <NoncurrentVersionTransition>
            <NoncurrentDays>30</NoncurrentDays>
            <StorageClass>GLACIER</StorageClass>
        </NoncurrentVersionTransition>
       <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>5</NewerNoncurrentVersions>
            <NoncurrentDays>365</NoncurrentDays>
       </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Exemple 7 : Suppression des marqueurs de suppression d'objet expiré

Un compartiment activé pour le contrôle de version a une version d'objet actuelle et une ou plusieurs versions anciennes pour chaque objet. Lorsque vous supprimez un objet, notez ce qui suit :

- Si vous ne spécifiez pas un ID de version dans votre demande de suppression, Amazon S3 ajoute un marqueur de suppression au lieu de supprimer l'objet. La version actuelle de l'objet devient ancienne et le marqueur de suppression devient la version actuelle.
- Si vous spécifiez un ID de version dans votre demande de suppression, Amazon S3 supprime la version de l'objet définitivement (aucun marqueur de suppression n'est créé).
- Un marqueur de suppression avec aucune version ancienne est appelé marqueur de suppression d'objet expiré.

Cet exemple illustre un scénario qui peut créer des marqueurs de suppression d'objet expiré dans votre compartiment et la façon dont vous pouvez utiliser une configuration de cycle de vie S3 pour indiquer à Amazon S3 de supprimer les marqueurs de suppression d'objet expiré.

Supposons que vous écriviez une configuration S3 Lifecycle qui utilise l'`NoncurrentVersionExpiration`action pour supprimer les versions non actuelles 30 jours

après leur disparition et qui conserve au maximum 10 versions non actuelles, comme indiqué dans l'exemple suivant.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

L'action `NoncurrentVersionExpiration`ne s'applique pas aux versions d'objets actuelles. Elle supprime uniquement les versions anciennes.

Pour les versions d'objet actuelles, les options suivantes vous permettent de gérer leur cycle de vie selon que les versions d'objet actuelles suivent un cycle de vie bien défini :

- Les versions d'objet actuelles suivent un cycle de vie bien défini.

  Dans ce cas, vous pouvez utiliser une configuration de cycle de vie S3 avec l'action `Expiration` pour indiquer à Amazon S3 de supprimer les versions actuelles, comme illustré dans l'exemple suivant.

  ```
  <LifecycleConfiguration>
      <Rule>
          ...
          <Expiration>
             <Days>60</Days>
          </Expiration>
          <NoncurrentVersionExpiration>
              <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
              <NoncurrentDays>30</NoncurrentDays>
          </NoncurrentVersionExpiration>
      </Rule>
  </LifecycleConfiguration>
  ```

  Dans cet exemple, Amazon S3 supprime les versions actuelles 60 jours après leur création en ajoutant un marqueur de suppression pour chaque version d'objet actuelle. Ainsi, la version

actuelle devient ancienne et le marqueur de suppression devient la version actuelle. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3.

> **Note**
>
> Vous ne pouvez pas spécifier à la fois une balise `Days` et `ExpiredObjectDeleteMarker` sur la même règle. Si vous spécifiez la balise `Days`, Amazon S3 effectue automatiquement le nettoyage `ExpiredObjectDeleteMarker` une fois que les marqueurs de suppression sont suffisamment anciens pour satisfaire aux critères d'âge. Pour nettoyer les marqueurs de suppression dès qu'ils deviennent la seule version, créez une règle distincte avec uniquement la balise `ExpiredObjectDeleteMarker`.

L'action `NoncurrentVersionExpiration` dans la même configuration de cycle de vie S3 supprime les objets anciens 30 jours après qu'ils sont devenus anciens. Ainsi, dans cet exemple, toutes les versions d'objet sont définitivement supprimées 90 jours après la création de l'objet. Bien que des marqueurs de suppression d'objet expiré soient créés au cours de ce processus, Amazon S3 détecte et supprime les marqueurs de suppression d'objet expiré à votre place.

- Les versions d'objet actuelles n'ont pas de cycle de vie bien défini.

  Dans ce cas, vous pouvez supprimer les objets manuellement si vous n'en avez pas besoin, ce qui créera un marqueur de suppression avec une ou plusieurs versions anciennes. Si la configuration de cycle de vie S3 avec l'action `NoncurrentVersionExpiration` supprime toutes les versions anciennes, vous avez à présent des marqueurs de suppression d'objet expiré.

  Plus précisément pour ce scénario, la configuration de cycle de vie S3 fournit une action `Expiration` que vous pouvez utiliser pour supprimer les marqueurs de suppression d'objet expiré.

```
<LifecycleConfiguration>
    <Rule>
       <ID>Rule 1</ID>
        <Filter>
          <Prefix>logs/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Expiration>
```

```
            <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

En définissant l'élément `ExpiredObjectDeleteMarker` sur `true` dans l'action `Expiration`, vous indiquez à Amazon S3 de supprimer les marqueurs de suppression d'objet expiré.

> **Note**
>
> Si vous spécifiez l'action de cycle de vie S3 `ExpiredObjectDeleteMarker`, la règle ne peut pas spécifier un filtre basé sur une balise.

Exemple 8 : Configuration du cycle de vie pour annuler des chargements partitionnés

Vous pouvez utiliser les opérations d'API REST de chargement partitionné Amazon S3 pour charger des objets volumineux en plusieurs parties. Pour en savoir plus sur le chargement partitionné, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.

En utilisant une configuration S3 Lifecycle, vous pouvez demander à Amazon S3 d'arrêter les téléchargements partitionnés incomplets (identifiés par le préfixe du nom de clé spécifié dans la règle) s'ils ne sont pas terminés dans un certain nombre de jours après leur lancement. Lorsque Amazon S3 interrompt un chargement partitionné, toutes les parties associées au chargement partitionné sont supprimées. Ce processus vous aide à contrôler vos coûts de stockage en garantissant que vous n'avez pas de chargements partitionnés incomplets avec des parties stockées dans Amazon S3.

> **Note**
>
> Si vous spécifiez l'action de cycle de vie S3 `AbortIncompleteMultipartUpload`, la règle ne peut pas spécifier un filtre basé sur une balise.

Voici un exemple de configuration de cycle de vie S3 qui spécifie une règle avec l'action `AbortIncompleteMultipartUpload`. Cette action indique à Amazon S3 d'arrêter les chargements partitionnés incomplets sept jours après leur lancement.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix>SomeKeyPrefix/</Prefix>
        </Filter>
        <Status>rule-status</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Exemple 9 : Configuration de cycle de vie à l'aide de règles basées sur la taille

Vous pouvez créer des règles qui transfèrent des objets uniquement en fonction de leur taille. Vous pouvez spécifier une taille minimale (`ObjectSizeGreaterThan`) ou une taille maximale (`ObjectSizeLessThan`), ou vous pouvez spécifier une plage de tailles d'objet en octets. Si vous utilisez plusieurs filtres, comme une règle de préfixe et de taille, vous devez les envelopper dans un élément `<And>`.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
       <And>
          <Prefix>tax/</Prefix>
          <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
       </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Si vous spécifiez une plage en utilisant à la fois les éléments `ObjectSizeGreaterThan` et `ObjectSizeLessThan`, la taille maximale des objets doit être supérieure à la taille minimale des objets. Si vous utilisez plusieurs filtres, vous devez les envelopper dans un élément `<And>`. L'exemple suivant montre comment spécifier des objets dans une plage comprise entre 500 octets et 64 000 octets. Lorsque vous spécifiez une plage, les `ObjectSizeLessThan` filtres `ObjectSizeGreaterThan` et excluent les valeurs spécifiées. Pour plus d'informations, consultez the section called “Élément de filtre”.

```
<LifecycleConfiguration>
    <Rule>
        ...
          <And>
             <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
             <ObjectSizeLessThan>64000</ObjectSizeLessThan>
          </And>
    </Rule>
</LifecycleConfiguration>
```

Vous pouvez également créer des règles pour faire expirer spécifiquement les anciens objets qui ne contiennent aucune donnée, y compris les anciens objets marqueurs de suppression, créés dans un compartiment doté de la gestion des versions. L'exemple suivant utilise l'action `NoncurrentVersionExpiration` pour supprimer les anciennes versions, 30 jours après qu'elles sont devenues anciennes, et pour retenir au plus 10 anciennes versions des objets. Il utilise également l'élément `ObjectSizeLessThan` pour filtrer uniquement les objets dépourvus de données.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Expire noncurrent with size less than 1 byte</ID>
    <Filter>
       <ObjectSizeLessThan>1</ObjectSizeLessThan>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
       <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
       <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Inventaire Simple Storage Service (Amazon S3)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Vous pouvez utiliser l'inventaire Amazon S3 pour vous aider à gérer votre stockage. Par exemple, vous pouvez l'utiliser pour contrôler et signaler le statut de réplication et de chiffrement de vos objets à des fins professionnelles, de conformité et d'obligations réglementaires. Vous pouvez également simplifier et accélérer les flux de travail et les tâches du big data à l'aide de l'inventaire Amazon S3, qui fournit une alternative planifiée aux opérations d'API `List` synchrones Amazon S3. L'inventaire Amazon S3 n'utilise pas les opérations d'API `List` pour auditer vos objets et n'affecte pas le taux de demande de votre compartiment.

L'inventaire Amazon S3 fournit des fichiers de sortie au format CSV (valeurs séparées par une virgule), ORC (Apache Optimized Row Columnar) ou Apache Parquet qui répertorient vos objets et leurs métadonnées correspondantes tous les jours ou toutes les semaines pour des objets ou un compartiment S3 avec un préfixe partagé (c'est-à-dire des objets dont le nom commence par une chaîne commune). Si vous avez configuré un inventaire hebdomadaire, un rapport est généré tous les dimanches (fuseau horaire UTC) après le rapport initial. Pour plus d'informations sur la tarification de l'inventaire Amazon S3, consultez Tarification Amazon S3.

Vous pouvez configurer plusieurs listes d'inventaire d'un compartiment. Lorsque vous configurez une liste d'inventaire, vous pouvez spécifier les éléments suivants :

- Quelles métadonnées d'objet inclure dans l'inventaire
- S'il faut répertorier toutes les versions de l'objet ou uniquement les versions actuelles
- Où stocker la sortie du fichier de liste d'inventaire

- S'il faut générer l'inventaire de manière quotidienne ou hebdomadaire
- S'il faut chiffrer le fichier de liste d'inventaire

Vous pouvez interroger l'inventaire Amazon S3 avec des requêtes SQL standard en utilisant Amazon Athena, Amazon Redshift Spectrum et d'autres outils tels que Presto, Apache Hive et Apache Spark. Pour plus d'informations sur l'utilisation d'Athena pour interroger vos fichiers d'inventaire, consultez the section called “Interrogation d'un inventaire avec Athena”.

Compartiments source et de destination

Le compartiment pour lequel l'inventaire répertorie les objets est appelé compartiment source. Le compartiment dans lequel le fichier de liste d'inventaire est stocké est appelé compartiment de destination.

Compartiment source

L'inventaire répertorie les objets qui sont stockés dans le compartiment source. Vous pouvez obtenir une liste d'inventaire d'un compartiment entier ou vous pouvez filtrer par liste par préfixe de nom de la clé d'objet.

Le compartiment source :

- Contient les objets qui sont répertoriés dans l'inventaire
- Contient la configuration de l'inventaire

Compartiment de destination

Les fichiers de liste d'inventaire Amazon S3 sont écrits dans le compartiment de destination. Vous pouvez spécifier un préfixe de destination dans la configuration de l'inventaire pour regrouper tous les fichiers de liste d'inventaire dans un emplacement commun au sein du compartiment de destination.

Le compartiment de destination :

- Contient les listes de fichiers d'inventaire.
- Contient les fichiers manifeste qui répertorient tous les fichiers de listes d'inventaire stockés dans le compartiment de destination. Pour plus d’informations, consultez Manifeste d'inventaire.
- Doit avoir une stratégie de compartiment pour donner à Amazon S3 l'autorisation de vérifier la propriété du compartiment et l'autorisation d'écrire des fichiers dans le compartiment.

- Doit se trouver dans le même compartiment Région AWS que le compartiment source.
- Peut être le même que le compartiment source.
- Peut appartenir à un compte Compte AWS différent du compte propriétaire du bucket source.

Liste d'inventaire Amazon S3

Un fichier de liste d'inventaire contient une liste des objets figurant dans le compartiment source et les métadonnées de chaque objet. Un fichier de liste d'inventaire est stocké dans le compartiment de destination sous l'un des formats suivants :

- Fichier CSV compressé avec GZIP
- En tant que fichier de colonne de ligne optimisée (ORC) par Apache compressé avec ZLIB
- En tant que fichier Apache Parquet compressé avec Snappy

> **Note**
>
> Le tri des objets n'est pas garanti dans les rapports d'inventaire Amazon S3.

Un fichier de liste d'inventaire contient une liste des objets figurant dans le compartiment source et les métadonnées de chaque objet figurant dans la liste :

- Nom du compartiment – Le nom du compartiment pour lequel l'inventaire est effectué.
- Nom de clé – Nom de la clé d'objet (ou clé) qui identifie de manière unique l'objet dans le compartiment. Lorsque vous utilisez le format de fichier CSV, le nom de clé est codé en URL et doit être décodé avant d'être utilisé.
- ID de version – ID de version de l'objet. Lorsque vous activez la gestion des versions sur un compartiment, Amazon S3 attribue un numéro de version aux objets qui sont ajoutés au compartiment. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3. (Ce champ n'est pas inclus si la liste est configurée uniquement pour la version actuelle des objets.)
- IsLatest— Défini sur `True` si l'objet est la version actuelle de l'objet. (Ce champ n'est pas inclus si la liste est configurée uniquement pour la version actuelle des objets.)
- Marqueur de suppression – Défini sur `True`, si l'objet est un marqueur de suppression. Pour plus d'informations, consultez Utilisation de la gestion des versions dans les compartiments S3. (Ce

champ est automatiquement ajouté à votre rapport si vous avez configuré celui-ci pour qu'il inclue toutes les versions de vos objets).

- Taille : la taille de l'objet en octets, à l'exclusion de la taille des chargements partitionnés incomplets, des métadonnées de l'objet et des marqueurs de suppression.
- Date de la dernière modification – Date de création de l'objet ou date de la dernière modification, la plus récente étant retenue.
- ETag : la balise d'entité (ETag) est un hachage de l'objet. ETag reflète les modifications uniquement appliquées au contenu d'un objet, pas à ses métadonnées. ETag peut être une valeur de hachage MD5 des données de l'objet. Cela dépend de la façon dont l'objet a été créé et de la manière dont il est chiffré.
- Classe de stockage : classe de stockage utilisée pour stocker l'objet. Défini sur `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR` ou `SNOW`. Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).
- Indicateur de chargement partitionné – Défini sur `True` si l'objet a été chargé dans un chargement partitionné. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.
- Statut de réplication – Défini sur `PENDING`, `COMPLETED`, `FAILED` ou `REPLICA`. Pour plus d'informations, consultez Obtention d'informations sur le statut de la réplication.
- État du chiffrement : état du chiffrement côté serveur, en fonction du type de clé de chiffrement utilisé : clé gérée par Amazon S3 (SSE-S3), clé AWS Key Management Service an AWS KMS() (SSE-KMS) ou clé fournie par le client (SSE-C). Définissez sur `SSE-S3`, `SSE-C`, `SSE-KMS` ou `NOT-SSE`. Le statut `NOT-SSE` signifie que l'objet n'est pas chiffré avec le chiffrement côté serveur. Pour plus d'informations, consultez Protection des données à l'aide du chiffrement.
- Date de fin de conservation du verrouillage d'objet S3 : date jusqu'à laquelle l'objet verrouillé ne peut pas être supprimé. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.
- Mode de rétention de verrouillage d'objet S3 : défini sur `Governance` ou `Compliance` pour les objets qui sont verrouillés. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.
- Statut de suspension juridique du verrouillage d'objet S3 : défini sur `On` si une suspension juridique a été appliquée à un objet. Sinon, elle est définie sur `Off`. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

- Niveau d'accès S3 Intelligent-Tiering : niveau d'accès (fréquent ou peu fréquent) de l'objet s'il est stocké dans S3 Intelligent-Tiering. Définissez sur `FREQUENT`, `INFREQUENT`, `ARCHIVE_INSTANT_ACCESS`, `ARCHIVE` ou `DEEP_ARCHIVE`. Pour plus d'informations, consultez Classe de stockage pour l'optimisation automatique des données avec des modèles d'accès inconnus ou irréguliers.
- Statut de clé de compartiment S3 – Défini sur `ENABLED` ou `DISABLED`. Indique si l'objet utilise une clé de compartiment S3 pour SSE-KMS. Pour plus d'informations, consultez Utilisation de clés de compartiment Amazon S3..
- Algorithme de total de contrôle : algorithme utilisé pour créer le total de contrôle de l'objet.
- Liste de contrôle d'accès aux objets : liste de contrôle d'accès (ACL) pour chaque objet qui définit le Comptes AWS ou les groupes autorisés à accéder à cet objet et le type d'accès accordé. Le champ Liste ACL d'objet est défini au format JSON. Un rapport d'inventaire S3 inclut les listes ACL associées aux objets dans votre compartiment source, même lorsque les listes ACL sont désactivées pour le compartiment. Pour plus d'informations, consultez Utiliser le champ Liste ACL d'objet et Présentation de la liste de contrôle d'accès (ACL).

> **Note**
>
> Le champ Liste ACL d'objet est défini au format JSON. Un rapport d'inventaire affiche la valeur du champ Liste ACL d'objet sous la forme d'une chaîne codée en base64. Supposons, par exemple, que vous disposiez du champ Liste ACL d'objet suivant au format JSON :
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```
>
> Le champ Liste ACL d'objet est codé et affiché sous la forme de la chaîne codée en base64 suivante :
>
> ```
> eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
> ```

Pour obtenir la valeur décodée au format JSON pour le champ Liste ACL d'objet, vous pouvez interroger ce champ dans Amazon Athena. Pour d'autres exemples de requête, consultez Interrogation d'un inventaire Amazon S3 avec Amazon Athena.

- Propriétaire de l'objet : le propriétaire de l'objet.

> **Note**
>
> Lorsqu'un objet est en fin de vie selon la configuration de son cycle de vie, Amazon S3 le place dans une file d'attente en vue de sa suppression et le supprime de manière asynchrone. Cependant, un certain retard est possible entre la date d'expiration et la date à laquelle Amazon S3 supprime l'objet. Le rapport d'inventaire inclut les objets qui ont expiré mais qui n'ont pas encore été supprimés. Pour plus d'informations sur les actions d'expiration dans le cycle de vie S3, consultez Objets en cours d'expiration.

Nous vous recommandons de créer une stratégie de cycle de vie qui supprime les anciennes listes d'inventaire. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.

L'autorisation `s3:PutInventoryConfiguration` permet à l'utilisateur de sélectionner tous les champs de métadonnées répertoriés précédemment pour chaque objet lors de la configuration d'une liste d'inventaire et de spécifier le compartiment de destination pour stocker l'inventaire. Un utilisateur disposant d'un accès en lecture aux objets du compartiment de destination peut accéder à tous les champs de métadonnées d'objets disponibles dans la liste d'inventaire. Pour restreindre l'accès à un rapport d'inventaire, consultez Accorder des autorisations pour l'inventaire S3 et les analyses S3..

Cohérence de l'inventaire

Tous vos objets peuvent ne pas apparaître dans chaque liste d'inventaire. La liste d'inventaire fournit une cohérence à terme pour des requêtes PUT (des objets nouveaux et de remplacement) et pour des requêtes `DELETE`. Chaque liste d'inventaire d'un compartiment est un instantané des articles du compartiment. Ces listes sont finalement cohérentes (c'est-à-dire qu'une liste peut ne pas inclure les objets récemment ajoutés ou supprimés).

Pour valider l'état d'un objet avant de prendre des mesures sur l'objet, nous recommandons d'effectuer une demande d'API REST `HeadObject` pour récupérer les métadonnées de l'objet ou de vérifier les propriétés de l'objet dans la console Amazon S3. Vous pouvez également vérifier

les métadonnées des objets avec le AWS CLI ou les AWS SDK. Pour plus d'informations, veuillez consulter `HeadObject` dans la Référence d'API Amazon Simple Storage Service.

Pour plus d'informations sur l'utilisation de l'inventaire Amazon S3, consultez les rubriques suivantes.

Rubriques

- Configuration d'Amazon S3 Inventory
- Configuration des notifications d'événements Amazon S3 pour l'achèvement de l'inventaire
- Localisation de votre liste d'inventaire
- Interrogation d'un inventaire Amazon S3 avec Amazon Athena
- Conversion de chaînes d'ID de version vides dans les rapports d'inventaire Amazon S3 en chaînes null
- Utiliser le champ Liste ACL d'objet

Configuration d'Amazon S3 Inventory

Amazon S3 Inventory fournit une liste de fichiers plats de vos objets et métadonnées, selon un calendrier que vous définissez. Vous pouvez utiliser S3 Inventory comme une autre solution planifiée de l'opération d'API `List` synchrone d'Amazon S3. S3 Inventory fournit des fichiers de sortie au format CSV (valeurs séparées par une virgule), ORC (Apache Optimized Row Columnar) ou Apache Parquet (Parquet) qui répertorient vos objets et leurs métadonnées correspondantes.

Vous pouvez configurer S3 Inventory afin de créer des listes d'inventaire de façon quotidienne ou hebdomadaire pour un compartiment S3 ou pour des objets qui partagent un préfixe (c'est-à-dire des objets dont le nom commence avec la même chaîne). Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).

Cette section explique comment configurer un inventaire, avec notamment des détails sur les compartiments source et de destination.

Rubriques

- Présentation
- Création d'une stratégie de compartiment de destination
- Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement
- Configuration de l'inventaire à l'aide de la console S3
- Utilisation de l'API REST pour travailler avec l'inventaire S3

Présentation

Amazon S3 Inventory vous aide à gérer votre stockage en créant des listes d'objets dans un compartiment S3 selon une planification définie. Vous pouvez configurer plusieurs listes d'inventaire d'un compartiment. Les listes d'inventaire sont publiées dans des fichiers CSV, ORC, ou Parquet au sein d'un compartiment de destination.

Le moyen le plus simple de configurer un inventaire est d'utiliser la console Amazon S3, mais vous pouvez également utiliser l'API REST AWS Command Line Interface (AWS CLI) ou les AWS kits SDK d'Amazon S3. La console effectue la première étape de la procédure suivante pour vous : ajouter une stratégie de compartiment au compartiment de destination.

Pour configure Amazon S3 Inventory pour un compartiment S3

1. Ajoutez une stratégie de compartiment pour le compartiment de destination.

   Vous devez créer une politique de compartiment sur le compartiment de destination qui autorise Amazon S3 à écrire des objets dans le compartiment à l'emplacement défini. Pour un exemple de stratégie, consultez Accorder des autorisations pour l'inventaire S3 et les analyses S3..

2. Configurez un inventaire pour répertorier les objets dans un compartiment source et publier la liste dans un compartiment de destination.

   Lorsque vous configurez une liste d'inventaire pour un compartiment source, vous spécifiez le compartiment de destination dans lequel vous voulez stocker la liste et la fréquence de génération de la liste (quotidienne ou hebdomadaire). Vous pouvez également configurer s'il faut répertorier toutes les versions d'objets ou uniquement les versions actuelles et les métadonnées d'objet à inclure.

   Certains champs de métadonnées d'objet dans les configurations des rapports d'inventaire S3 sont facultatifs, ce qui signifie qu'ils sont disponibles par défaut mais qu'ils peuvent être restreints lorsque vous accordez l'`s3:PutInventoryConfiguration`autorisation à un utilisateur. Vous pouvez contrôler si les utilisateurs peuvent inclure ces champs de métadonnées facultatifs dans leurs rapports à l'aide de la clé de `s3:InventoryAccessibleOptionalFields` condition.

   Pour plus d'informations sur les champs de métadonnées facultatifs disponibles dans S3 Inventory, consultez OptionalFieldsle manuel Amazon Simple Storage Service API Reference. Pour plus d'informations sur la restriction de l'accès à certains champs de métadonnées facultatifs dans une configuration d'inventaire, consultezCréation de la configuration du rapport d'inventaire Control S3.

Vous pouvez spécifier que le fichier de liste d'inventaire soit chiffré en utilisant le chiffrement côté serveur avec une clé gérée Amazon S3 (SSE-S3) ou une clé gérée par le client AWS Key Management Service (SSE-KMS AWS KMS).

> **Note**
>
> Le Clé gérée par AWS (`aws/s3`) n'est pas pris en charge pour le chiffrement SSE-KMS avec S3 Inventory.

Pour en savoir plus sur SSE-S3 et SSE-KMS, consultez Protection des données avec le chiffrement côté serveur. Si vous prévoyez d'utiliser le chiffrement SSE-KMS, consultez l'étape 3.

- Pour plus d'informations sur l'utilisation de la console pour configurer une liste d'inventaire, consultez Configuration de l'inventaire à l'aide de la console S3.
- Pour utiliser l'API Amazon S3 afin de configurer une liste d'inventaire, utilisez l'opération d'PutBucketInventoryConfigurationAPI REST AWS CLI ou l'équivalent AWS des SDK.

3. Pour chiffrer le fichier de liste d'inventaire avec SSE-KMS, autorisez Amazon S3 à utiliser la AWS KMS key.

   Vous pouvez configurer le chiffrement du fichier de liste d'inventaire à l'aide de la console Amazon S3, de l'API REST Amazon S3 ou AWS des kits SDK. AWS CLI Quelle que soit la méthode que vous choisissez, vous devez autoriser Amazon S3 à utiliser la clé gérée par le client pour chiffrer le fichier d'inventaire. Vous accordez l'autorisation à Amazon S3 en modifiant la politique de la clé gérée par le client que vous souhaitez utiliser pour chiffrer le fichier d'inventaire. Pour plus d'informations, consultez Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement.

   Le compartiment de destination qui stocke le fichier de liste d'inventaire peut appartenir à un autre Compte AWS que le compte propriétaire du compartiment source. Si vous utilisez le chiffrement SSE-KMS pour les opérations entre comptes d'Amazon S3 Inventory, nous vous recommandons d'utiliser un ARN de clé KMS complet lorsque vous configurez l'inventaire S3. Pour plus d'informations, consultez Utilisation du chiffrement SSE-KMS pour les opérations intercomptes et ServerSideEncryptionByDefault dans la référence de l'API Amazon Simple Storage Service.

Création d'une stratégie de compartiment de destination

Si vous créez votre configuration d'inventaire via la console Amazon S3, Amazon S3 crée automatiquement une politique de compartiment sur le compartiment de destination qui accorde une autorisation d'écriture Amazon S3 au compartiment. Toutefois, si vous créez votre configuration d'inventaire via les AWS CLI AWS SDK ou l'API REST Amazon S3, vous devez ajouter manuellement une politique de compartiment sur le compartiment de destination. Pour plus d'informations, consultez Accorder des autorisations pour l'inventaire S3 et les analyses S3.. La politique du compartiment de destination de l'inventaire S3 permet à Amazon S3 d'écrire des données pour les rapports d'inventaire dans le compartiment.

Si une erreur se produit lorsque vous tentez de créer la stratégie de compartiment, des instructions s'affichent pour vous indiquer comment la résoudre. Par exemple, si vous choisissez un compartiment de destination dans un autre Compte AWS et que vous n'êtes pas autorisé à lire et à écrire dans la politique du compartiment, un message d'erreur s'affiche.

Dans ce cas, le propriétaire du compartiment de destination doit ajouter la politique du compartiment au compartiment de destination. Si la politique n'est pas ajoutée au compartiment de destination, vous ne recevez pas de rapport d'inventaire, car Amazon S3 n'est pas autorisé à écrire dans le compartiment de destination. Si le compartiment source est détenu par un compte autre que celui de l'utilisateur actuel, l'ID de compte correct du propriétaire du compartiment source doit être remplacé dans la politique.

Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement

Pour autoriser Amazon S3 à utiliser votre clé gérée par le client AWS Key Management Service (AWS KMS) pour le chiffrement côté serveur, vous devez utiliser une politique de clé. Pour mettre à jour votre stratégie de clé et pouvoir utiliser votre clé gérée par le client, suivez la procédure suivante.

Pour autoriser Amazon S3 à chiffrer à l'aide de votre clé gérée par le client

1. À l'aide du Compte AWS propriétaire de la clé gérée par le client, connectez-vous au AWS Management Console.
2. Ouvrez la AWS KMS console à l'adresse https://console.aws.amazon.com/kms.
3. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
4. Dans le panneau de navigation de gauche, choisissez Clés gérées par le client.

5. Sous Clés gérées par le client, choisissez la clé gérée par le client que vous souhaitez utiliser pour chiffrer vos fichiers d'inventaire.
6. Dans la section Key policy (Politique de clé), sélectionnez Switch to policy view (Passer à la vue de politique).
7. Pour mettre à jour la politique de clé, choisissez Modifier.
8. Sous Modifier la stratégie de clé, ajoutez les lignes suivantes à la stratégie de clé existante. Pour *source-account-id* et *example-s3-source-bucket*, fournissez les valeurs appropriées pour votre cas d'utilisation.

```
{
    "Sid": "Allow Amazon S3 use of the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "Service": "s3.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition":{
      "StringEquals":{
         "aws:SourceAccount":"source-account-id"
     },
      "ArnLike":{
        "aws:SourceARN": "arn:aws:s3:::example-s3-source-bucket"
     }
   }
}
```

9. Choisissez Enregistrer les modifications.

Pour en savoir plus amples sur la création de clés gérées par le client et l'utilisation de politiques de clé, cliquez sur les liens suivants dans le guide du développeur AWS Key Management Service :

- Gestion des clés
- Politiques clés en AWS KMS

Configuration de l'inventaire à l'aide de la console S3

Suivez ces instructions pour configurer l'inventaire à l'aide de la console S3.

> **Note**
>
> La distribution du premier rapport d'inventaire pour Amazon S3 peut prendre jusqu'à 48 heures.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments. Dans la liste Compartiments, choisissez le nom du compartiment pour lequel vous souhaitez configurer l'inventaire Amazon S3.
3. Choisissez l'onglet Gestion.
4. Sous Configurations d'inventaire, choisissez Créer une configuration d'inventaire.
5. Dans Nom de la configuration d'inventaire, saisissez un nom.
6. Pour Portée de l'inventaire, procédez comme suit :
   - Entrez un préfixe facultatif.
   - Choisissez les versions d'objet à inclure, soit Versions actuelles uniquement, soit Inclure toutes les versions.
7. Sous Report details (Détails du rapport), choisissez l'emplacement du Compte AWS où vous souhaitez enregistrer les rapports : This account (Ce compte) ou A different account (Un compte différent).
8. Sous Destination, choisissez le compartiment de destination dans lequel vous souhaitez que les rapports d'inventaire soient enregistrés.

   Le compartiment de destination doit se trouver dans le même compartiment Région AWS que celui pour lequel vous configurez l'inventaire. Le compartiment de destination peut se trouver dans un autre Compte AWS. Lorsque vous spécifiez le compartiment de destination, vous pouvez également inclure un préfixe facultatif pour regrouper vos rapports d'inventaire.

   Sous le champ de compartiment Destination vous voyez la déclaration Autorisation de compartiment de destination qui est ajoutée à la stratégie de compartiment de destination pour

permettre à Amazon S3 de placer des données dans ce compartiment. Pour plus d'informations, consultez Création d'une stratégie de compartiment de destination.

9. Sous Fréquence, choisissez la fréquence à laquelle le rapport sera généré : Quotidien ou Hebdomadaire.
10. Dans Format de sortie, choisissez l'un des formats suivants pour le rapport :
    - CSV : si vous prévoyez d'utiliser ce rapport d'inventaire avec S3 Batch Operations ou si vous souhaitez analyser ce rapport dans un autre outil, tel que Microsoft Excel, choisissez CSV.
    - Apache ORC
    - Apache Parquet
11. Sous Status (Statut), choisissez Enable (Activer) ou Disable (Désactiver).
12. Pour configurer le chiffrement côté serveur, sous Chiffrement des rapports d'inventaire, procédez comme suit :
    a. Sous Chiffrement côté serveur, choisissez Ne pas spécifier de clé de chiffrement ou Spécifier une clé de chiffrement pour chiffrer les données.
        - Pour conserver les paramètres du compartiment pour le chiffrement côté serveur par défaut des objets lors de leur stockage dans Amazon S3, choisissez Ne pas spécifier de clé de chiffrement. Tant que les clés de compartiment S3 sont activées pour la destination du compartiment, l'opération de copie applique une clé de compartiment S3 au compartiment de destination.

          > **Note**
          >
          > Si la politique de compartiment pour la destination spécifiée exige que les objets soient chiffrés avant de les stocker dans Amazon S3, vous devez choisir Spécifier une clé de chiffrement. Sinon, la copie des objets vers la destination échouera.

        - Pour chiffrer des objets avant de les stocker dans Amazon S3, choisissez Spécifier une clé de chiffrement.
    b. Si vous avez choisi Spécifier une clé de chiffrement, sous Type de chiffrement, vous devez choisir la clé gérée Amazon S3 (SSE-S3) ou la AWS Key Management Service clé (SSE-KMS).

        SSE-S3 utilise l'un des chiffrements par bloc les plus puissants qui existent, Advanced Encryption Standard à 256 bits (AES-256) pour chiffrer chaque objet. SSE-KMS vous

permet de mieux contrôler votre clé. Pour en savoir plus sur SSE-S3, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3). Pour en savoir plus sur SSE-KMS, consultez Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

> **Note**
>
> Pour chiffrer le fichier de liste d'inventaire avec SSE-KMS, vous devez autoriser Amazon S3 à utiliser la clé gérée par le client. Pour obtenir des instructions, consultez Octroyer à Amazon S3 l'autorisation d'utiliser vos clés KMS pour le chiffrement.

c. Si vous avez choisi AWS Key Management Service clé (SSE-KMS), sous AWS KMS key, vous pouvez spécifier votre AWS KMS clé à l'aide de l'une des options suivantes.

> **Note**
>
> Si le compartiment de destination qui stocke le fichier de liste d'inventaire appartient à un autre Compte AWS utilisateur, assurez-vous d'utiliser un ARN de clé KMS complet pour spécifier votre clé KMS.

- Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS clés, puis choisissez une clé KMS de chiffrement symétrique dans la liste des clés disponibles. Assurez-vous que la clé KMS se trouve dans la même région que votre compartiment.

  > **Note**
  >
  > La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans la liste. Cependant, le Clé gérée par AWS (`aws/s3`) n'est pas pris en charge pour le chiffrement SSE-KMS avec S3 Inventory.

- Pour saisir l'ARN de la clé KMS, choisissez Enter AWS KMS key ARN, puis saisissez l'ARN de votre clé KMS dans le champ qui apparaît.
- Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

13. Pour Champs de métadonnées supplémentaires, sélectionnez un ou plusieurs des éléments suivants à ajouter au rapport d'inventaire :

    - Taille : la taille de l'objet en octets, à l'exclusion de la taille des chargements partitionnés incomplets, des métadonnées de l'objet et des marqueurs de suppression.
    - Date de la dernière modification – Date de création de l'objet ou date de la dernière modification, la plus récente étant retenue.
    - Multipart upload (Chargement partitionné) – Spécifie que l'objet a été chargé dans un chargement partitionné. Pour de plus amples informations, veuillez consulter Chargement et copie d'objets à l'aide d'un chargement partitionné.
    - Replication status (Statut de réplication) – Statut de réplication de l'objet. Pour plus d'informations, consultez Obtention d'informations sur le statut de la réplication.
    - Statut de chiffrement : chiffrement côté serveur utilisé pour chiffrer l'objet. Pour plus d'informations, consultez Protection des données avec le chiffrement côté serveur.
    - État de la clé du compartiment : indique si une clé au niveau du compartiment générée par AWS KMS s'applique à l'objet. Pour plus d'informations, consultez Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3.
    - Liste de contrôle d'accès aux objets : liste de contrôle d'accès (ACL) pour chaque objet qui définit le Comptes AWS ou les groupes autorisés à accéder à cet objet et le type d'accès accordé. Pour plus d'informations sur ce champ, consultez Utiliser le champ Liste ACL d'objet. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).
    - Propriétaire de l'objet : le propriétaire de l'objet.
    - Classe de stockage : classe de stockage utilisée pour stocker l'objet.
    - Niveau d'accès Intelligent-Tiering : niveau d'accès (fréquent ou peu fréquent) de l'objet si celui-ci est stocké dans la classe de stockage S3 Intelligent-Tiering. Pour plus d'informations, consultez Classe de stockage pour l'optimisation automatique des données avec des modèles d'accès inconnus ou irréguliers.
    - ETag : la balise d'entité (ETag) est un hachage de l'objet. ETag reflète les modifications uniquement appliquées au contenu d'un objet, pas à ses métadonnées. ETag peut ou ne peut pas être une valeur de hachage MD5 des données de l'objet. Cela dépend de la façon dont l'objet a été créé et de la manière dont il est chiffré. Pour plus d'informations, veuillez consulter Object dans la Référence d'API Amazon Simple Storage Service.
    - Algorithme de total de contrôle : algorithme utilisé pour créer le total de contrôle de l'objet.

- Configurations du verrouillage de tous les objets : statut du verrouillage des objets, y compris les paramètres suivants :
  - Verrouillage d'objet : mode de rétention : niveau de protection appliqué à l'objet, Gouvernance ou Conformité.
  - Verrouillage d'objet : conserver jusqu'à la date : date jusqu'à laquelle l'objet verrouillé ne peut pas être supprimé.
  - Verrouillage d'objet : statut de la conservation à des fins juridiques : statut de la conservation à des fins juridiques de l'objet verrouillé.

  Pour de plus amples informations sur la fonctionnalité de verrouillage des objets S3, veuillez consulter Fonctionnement du verrouillage d'objets S3.

Pour plus d'informations sur le contenu d'un rapport d'inventaire, veuillez consulter Liste d'inventaire Amazon S3.

Pour plus d'informations sur la restriction de l'accès à certains champs de métadonnées facultatifs dans une configuration d'inventaire, consultezCréation de la configuration du rapport d'inventaire Control S3.

14. Choisissez Créer.

Lorsqu'une liste d'inventaire est publiée, vous pouvez interroger le fichier de liste d'inventaire avec Amazon S3 Select. Pour plus d'informations sur la façon de localiser votre liste d'inventaire et d'interroger le fichier de liste d'inventaire avec Amazon S3 Select, consultez Localisation de votre liste d'inventaire.

Utilisation de l'API REST pour travailler avec l'inventaire S3

Voici les opérations REST que vous pouvez utiliser pour travailler avec Amazon S3 Inventory.

- DeleteBucketInventoryConfiguration
- GetBucketInventoryConfiguration
- ListBucketInventoryConfigurations
- PutBucketInventoryConfiguration

Configuration des notifications d'événements Amazon S3 pour l'achèvement de l'inventaire

Vous pouvez configurer une notification d'événement Amazon S3 afin de recevoir une notification lorsque le fichier du total de contrôle manifeste est créé, ce qui indique qu'une liste d'inventaire a été ajoutée au compartiment de destination. Le manifeste est une up-to-date liste de toutes les listes d'inventaire du site de destination.

Amazon S3 peut publier des événements dans une rubrique Amazon Simple Notification Service (Amazon SNS), une file d'attente Amazon Simple Queue Service (Amazon SQS) ou une fonction AWS Lambda . Pour plus d'informations, consultez Notifications d'événements Amazon S3.

La configuration de notification suivante définit que tous les fichiers `manifest.checksum` nouvellement ajoutés au compartiment de destination sont traités par l'opération AWS Lambda `cloud-function-list-write`.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>destination-prefix/source-bucket</Value>
              </FilterRule>
              <FilterRule>
                  <Name>suffix</Name>
                  <Value>checksum</Value>
              </FilterRule>
          </S3Key>
      </Filter>
      <Cloudcode>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</
Cloudcode>
      <Event>s3:ObjectCreated:*</Event>
  </QueueConfiguration>
  </NotificationConfiguration>
```

Pour plus d'informations, consultez la section Utilisation AWS Lambda avec Amazon S3 dans le manuel du AWS Lambda développeur.

Localisation de votre liste d'inventaire

Lorsqu'une liste d'inventaire est publiée, les fichiers manifestes sont publiés dans l'emplacement suivant dans le compartiment de destination.

```
destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.json
 destination-prefix/source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.checksum
 destination-prefix/source-bucket/config-ID/hive/dt=YYYY-MM-DD-HH-MM/symlink.txt
```

- *destination-prefix* est le préfixe de nom de la clé d'objet défini de manière facultative dans la configuration de l'inventaire. Vous pouvez utiliser ce préfixe pour regrouper tous les fichiers de liste d'inventaire dans un emplacement commun au sein du compartiment de destination.
- *source-bucket* est le compartiment source pour lequel la liste d'inventaire est effectuée. Le nom du compartiment source est ajouté pour éviter les collisions lorsque plusieurs rapports d'inventaire de différents compartiments source sont envoyés dans le même compartiment de destination.
- *config-ID* est ajouté pour éviter les collisions avec plusieurs rapports d'inventaire du même compartiment source qui sont envoyés dans le même compartiment de destination. Le *config-ID* provient de la configuration de rapport d'inventaire et est le nom du rapport défini lors de la configuration.
- *YYYY-MM-DDTHH-MMZ* est l'horodatage composé de l'heure de début et de la date à laquelle la génération de rapport d'inventaire a commencé à analyser le compartiment, par exemple, `2016-11-06T21-32Z`.
- `manifest.json` est le fichier manifeste.
- `manifest.checksum` est le hachage MD5 du contenu du fichier `manifest.json`.
- `symlink.txt` est le fichier manifeste compatible avec Apache Hive.

Les listes d'inventaire sont publiées sur une base quotidienne ou hebdomadaire dans l'emplacement suivant dans le compartiment de destination.

```
destination-prefix/source-bucket/config-ID/data/example-file-name.csv.gz
...
destination-prefix/source-bucket/config-ID/data/example-file-name-1.csv.gz
```

- *destination-prefix* est le préfixe de nom de la clé d'objet défini de manière facultative dans la configuration de l'inventaire. Vous pouvez utiliser ce préfixe pour regrouper tous les fichiers de liste d'inventaire dans un emplacement commun au sein du compartiment de destination.

- *source-bucket* est le compartiment source pour lequel la liste d'inventaire est effectuée. Le nom du compartiment source est ajouté pour éviter les collisions lorsque plusieurs rapports d'inventaire de différents compartiments source sont envoyés dans le même compartiment de destination.
- *example-file-name*`.csv.gz` est l'un des fichiers de l'inventaire CSV. Les noms d'inventaire ORC se terminent par l'extension de nom de fichier `.orc` et les noms d'inventaire Parquet se terminent par l'extension de nom de fichier `.parquet`.

  Vous pouvez rechercher un fichier de liste d'inventaire avec Amazon S3 Select. *Dans la console Amazon S3, choisissez le nom de la liste d'inventaire (par exemple, destination-prefixe/source-bucket /config-ID /data/ .csv.gz). example-file-name* Choisissez Actions d'objet, puis Requête avec S3 Select. Pour un exemple d'utilisation d'une fonction d'agrégation S3 Select pour interroger un fichier de liste d'inventaire, consultez Exemple de SUM.

Manifeste d'inventaire

Les fichiers manifestes `manifest.json` et `symlink.txt` décrivent l'emplacement où les fichiers d'inventaire sont situés. Lorsqu'une nouvelle liste d'inventaire est fournie, elle est accompagnée d'un nouvel ensemble de fichiers manifestes. Ces fichiers peuvent s'écraser les uns les autres. Dans les compartiments soumis au contrôle de version, Amazon S3 crée de nouvelles versions des fichiers de manifeste.

Chaque manifeste contenu dans le fichier `manifest.json` fournit des métadonnées et d'autres informations de base sur un inventaire. Les informations collectées sont les suivantes :

- Nom de compartiment source
- Nom du compartiment de destination
- Version de l'inventaire
- Horodatage de création, au format de date d'époque, composé de l'heure de début et de la date à laquelle le processus de génération de rapport d'inventaire a commencé à analyser le compartiment
- Format et schéma des fichiers d'inventaire
- Liste des fichiers d'inventaire se trouvant dans le compartiment de destination

Chaque fois qu'un fichier `manifest.json` est écrit, il est accompagné d'un fichier `manifest.checksum` qui est le hachage MD5 du contenu du fichier `manifest.json`.

Example Manifeste d'inventaire dans un fichier **`manifest.json`**

Voici des exemples de manifeste d'inventaire dans un fichier `manifest.json` pour les inventaires au format CSV, ORC et Parquet.

CSV

Voici un exemple de manifeste dans un fichier `manifest.json` pour un inventaire au format CSV.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-inventory-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
 Size, LastModifiedDate, ETag, StorageClass, IsMultipartUploaded,
 ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode,
 ObjectLockLegalHoldStatus, IntelligentTieringAccessTier, BucketKeyStatus,
 ChecksumAlgorithm, ObjectAccessControlList, ObjectOwner",
    "files": [
        {
            "key": "Inventory/example-source-bucket/2016-11-06T21-32Z/
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",
            "size": 2147483647,
            "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"
        }
    ]
}
```

ORC

Voici un exemple de manifeste dans un fichier `manifest.json` pour un inventaire au format ORC.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "ORC",
```

```
    "fileSchema":
 "struct<bucket:string,key:string,version_id:string,is_latest:boolean,is_delete_marker:boole
    "files": [
        {
            "key": "inventory/example-source-bucket/data/
d794c570-95bb-4271-9128-26023c8b4900.orc",
            "size": 56291,
            "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"
        }
    ]
}
```

Parquet

Voici un exemple de manifeste dans un fichier `manifest.json` pour un inventaire au format Parquet.

```
{
    "sourceBucket": "example-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "Parquet",
    "fileSchema": "message s3.inventory { required binary bucket (UTF8);
 required binary key (UTF8); optional binary version_id (UTF8); optional boolean
 is_latest; optional boolean is_delete_marker; optional int64 size; optional
 int64 last_modified_date (TIMESTAMP_MILLIS); optional binary e_tag (UTF8);
 optional binary storage_class (UTF8); optional boolean is_multipart_uploaded;
 optional binary replication_status (UTF8); optional binary encryption_status
 (UTF8); optional int64 object_lock_retain_until_date (TIMESTAMP_MILLIS); optional
 binary object_lock_mode (UTF8); optional binary object_lock_legal_hold_status
 (UTF8); optional binary intelligent_tiering_access_tier (UTF8); optional binary
 bucket_key_status (UTF8); optional binary checksum_algorithm (UTF8); optional
 binary object_access_control_list (UTF8); optional binary object_owner (UTF8);}",
    "files": [
        {
           "key": "inventory/example-source-bucket/data/
d754c470-85bb-4255-9218-47023c8b4910.parquet",
            "size": 56291,
            "MD5checksum": "5825f2e18e1695c2d030b9f6eexample"
        }
    ]
}
```

Le fichier `symlink.txt` est un fichier manifeste compatible avec Apache Hive qui permet à de découvrir automatiquement les fichiers d'inventaire et leurs fichiers de données associés. Le manifeste compatible avec Hive fonctionne avec les services Athena et Amazon Redshift Spectrum compatibles avec Hive. Il fonctionne également avec les applications compatibles avec Hive, notamment Presto, Apache Hive, Apache Spark et bien d'autres applications.

> **Important**
>
> Le fichier manifeste compatible avec `symlink.txt` Apache Hive ne fonctionne pas avec AWS Glue actuellement.
> La lecture du fichier `symlink.txt` avec Apache Hive et Apache Spark n'est pas prise en charge par les fichiers d'inventaire au format ORC ou Parquet.

Interrogation d'un inventaire Amazon S3 avec Amazon Athena

Vous pouvez interroger les fichiers de l'inventaire Amazon S3 en utilisant des requêtes SQL standard à l'aide d'Amazon Athena dans toutes les Régions où Athena est disponible. Pour vérifier la disponibilité de la Région AWS, veuillez consulter la Table Région AWS.

Athena peut interroger les fichiers d'inventaire Amazon S3 dans des colonnes de ligne optimisées (ORC) Apache, Apache Parquet ou au format CSV (valeurs séparées par des virgules). Lorsque vous utilisez Athena pour interroger les fichiers d'inventaire, nous vous recommandons d'utiliser des fichiers d'inventaire au format ORC ou Parquet. Les formats ORC et Parquet offrent des performances de requête plus rapides et des coûts de requête réduits. Les formats ORC et Parquet sont des formats auto-descriptifs en colonnes avec prise en charge du type, conçus pour Apache Hadoop. Le format en colonnes permet de lire, décompresser et traiter uniquement les colonnes nécessaires pour traiter la requête actuelle. Les formats ORC et Parquet pour l'inventaire Amazon S3 sont disponibles dans toutes les Régions AWS.

Pour utiliser Athena pour interroger les fichiers d'inventaire Amazon S3

1. Créez une table Athéna. Pour de plus amples informations sur la création d'une table, veuillez consulter Création de tables dans Athena dans le Guide de l'utilisateur Amazon Athena.
2. Créez votre requête à l'aide de l'un des exemples de modèles de requête suivants, selon que vous interrogez un rapport d'inventaire au format ORC, Parquet ou CSV.

- Lorsque vous utilisez Athena pour interroger un rapport d'inventaire au format CSV, utilisez l'exemple de requête suivant en tant que modèle.

  L'exemple de requête suivant comprend tous les champs facultatifs dans le rapport d'inventaire au format ORC.

  Pour utiliser cet exemple de requête, procédez comme suit :

  - Remplacez *your_table_name* par le nom de la table Athena que vous avez créée.
  - Supprimez les champs facultatifs que vous n'avez pas choisis pour votre inventaire afin que la requête corresponde aux champs sélectionnés pour celui-ci.
  - Remplacez le nom de compartiment et l'emplacement d'inventaire (l'ID de configuration) suivants en fonction de votre configuration.

    `s3://`*DOC-EXAMPLE-BUCKET*`/`*config-ID*`/hive/`

  - Remplacez la date *2022-01-01-00-00* sous `projection.dt.range` par le premier jour de la plage horaire au cours de laquelle vous partitionnez les données dans Athena. Pour plus d'informations, consultez Partitionnement de données dans Athena.

```
CREATE EXTERNAL TABLE your_table_name(
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size bigint,
         last_modified_date timestamp,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
         object_lock_retain_until_date bigint,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
```

```
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );
```

- Lorsque vous utilisez Athena pour interroger un rapport d'inventaire au format Parquet, utilisez l'exemple de requête pour un rapport au format ORC. Toutefois, utilisez le SerDe Parquet suivant à la place du SerDe ORC dans la déclaration `ROW FORMAT SERDE`.

```
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
```

- Lorsque vous utilisez Athena pour interroger un rapport d'inventaire au format CSV, utilisez l'exemple de requête suivant en tant que modèle.

  L'exemple de requête suivant comprend tous les champs facultatifs dans le rapport d'inventaire au format CSV.

  Pour utiliser cet exemple de requête, procédez comme suit :

  - Remplacez *`your_table_name`* par le nom de la table Athena que vous avez créée.
  - Supprimez les champs facultatifs que vous n'avez pas choisis pour votre inventaire afin que la requête corresponde aux champs sélectionnés pour celui-ci.
  - Remplacez le nom de compartiment et l'emplacement d'inventaire (l'ID de configuration) suivants en fonction de votre configuration.

    `s3://`*`DOC-EXAMPLE-BUCKET`*`/`*`config-ID`*`/hive/`

  - Remplacez la date *`2022-01-01-00-00`* sous `projection.dt.range` par le premier jour de la plage horaire au cours de laquelle vous partitionnez les données dans Athena. Pour plus d'informations, consultez Partitionnement de données dans Athena.

```
CREATE EXTERNAL TABLE your_table_name(
```

```
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size string,
         last_modified_date string,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
         object_lock_retain_until_date string,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );
```

3. Vous pouvez désormais exécuter différentes requêtes sur votre inventaire, comme le montrent les exemples suivants. Remplacez chaque *user input placeholder* par vos propres informations.

```
# Get a list of the latest inventory report dates available.
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;
```

```
# Get the encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-
MM' GROUP BY encryption_status;

# Get the encryption status for inventory report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
 encryption_status;
```

Lorsque vous configurez l'inventaire S3 pour ajouter le champ Liste de contrôle d'accès d'objet (Liste ACL d'objet) à un rapport d'inventaire, le rapport affiche la valeur du champ Liste ACL d'objet sous la forme d'une chaîne codée en base64. Pour obtenir la valeur décodée au format JSON pour le champ Liste ACL d'objet, vous pouvez interroger ce champ avec Athena. Consultez les exemples de requête suivants. Pour plus d'informations sur le champ Liste ACL d'objet, consultez Utiliser le champ Liste ACL d'objet.

```
# Get the S3 keys that have Object ACL grants with public access.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'uri') = 'http://acs.amazonaws.com/groups/global/AllUsers'
```

```
# Get the S3 keys that have Object ACL grantees in addition to the object owner.
WITH grants AS
    (SELECT key,
    from_utf8(from_base64(object_access_control_list)) AS
 object_access_control_list,
         object_owner,
         CAST(json_extract(from_utf8(from_base64(object_access_control_list)),
         '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))) AS grants_array
    FROM your_table_name)
```

```
SELECT key,
       grant,
       objectowner
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE cardinality(grants_array) > 1 AND element_at(grant, 'canonicalId') !=
 object_owner;
```

```
# Get the S3 keys with READ permission that is granted in the Object ACL.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'permission') = 'READ';
```

```
# Get the S3 keys that have Object ACL grants to a specific canonical user ID.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'canonicalId') = 'user-canonical-id';
```

```
# Get the number of grantees on the Object ACL.
SELECT key,
```

```
       object_access_control_list,
       json_array_length(json_extract(object_access_control_list,'$.grants')) AS
 grants_count
FROM your_table_name;
```

Pour plus d'informations sur l'utilisation d'Athena, consultez le Guide de l'utilisateur Amazon Athena.

Conversion de chaînes d'ID de version vides dans les rapports d'inventaire Amazon S3 en chaînes null

> **Note**
>
> La procédure suivante s'applique uniquement aux rapports d'inventaire Amazon S3 qui incluent toutes les versions, et uniquement si les rapports « toutes les versions » sont utilisés comme manifestes pour S3 Batch Operations sur des compartiments sur lesquels la gestion des versions S3 est activée. Il n'est pas nécessaire de convertir des chaînes pour les rapports d'inventaire S3 qui spécifient uniquement la version actuelle.

Vous pouvez utiliser les rapports d'inventaire S3 en tant que manifestes pour S3 Batch Operations. Toutefois, si la gestion des versions S3 est activée sur un compartiment, les rapports d'inventaire S3 qui incluent toutes les versions marquent tous les objets versionnés NULL avec des chaînes vides dans le champ d'ID de version. Si un rapport d'inventaire inclut tous les ID de version d'objet, Batch Operations reconnaît les chaînes `null` comme des ID de version, mais pas comme des chaînes vides.

Si une tâche S3 Batch Operations utilise un rapport d'inventaire S3 « toutes les versions » comme manifeste, il échoue toutes les tâches sur les objets dont la chaîne est vide dans le champ d'ID de version. Pour convertir des chaînes vides dans le champ d'ID de version du rapport d'inventaire S3 en chaînes `null` pour Batch Operations, suivez la procédure suivante.

Mise à jour d'un rapport d'inventaire Amazon S3 pour l'utiliser avec Batch Operations

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Accédez à votre rapport d'inventaire S3. Le rapport d'inventaire se trouve dans le compartiment de destination que vous avez spécifié lors de la configuration de votre rapport d'inventaire. Pour

de plus amples informations sur la manière de trouver les rapports d'inventaire, veuillez consulter Localisation de votre liste d'inventaire.

a. Choisissez le compartiment de destination.

b. Choisissez le dossier. Le dossier porte le nom du compartiment source d'origine.

c. Choisissez le dossier nommé d'après la configuration d'inventaire.

d. Cochez la case en regard du dossier nommé hive. En haut de la page, choisissez Copy S3 URI (Copier l'URI S3) pour copier l'URI S3 du dossier.

3. Ouvrez la console Amazon Athena à l'adresse https://console.aws.amazon.com/athena/.

4. Dans l'éditeur de requête, choisissez Settings (Paramètres), puis Manage (Gérer). Sur la page Manage settings (Gérer les paramètres), pour Location of query result (Emplacement des résultats de la requête), choisissez un compartiment S3 dans lequel stocker les résultats de votre requête.

5. Dans l'éditeur de requête, créez une table Athena pour contenir les données du rapport d'inventaire à l'aide de la commande suivante. Remplacez *table_name* par un nom de votre choix, et dans la clause `LOCATION`, insérez l'URI S3 que vous avez copié précédemment. Choisissez ensuite Run (Exécuter) pour exécuter la requête.

```
CREATE EXTERNAL TABLE table_name(bucket string, key string,
 version_id string) PARTITIONED BY (dt string)ROW FORMAT SERDE
 'org.apache.hadoop.hive.serde2.OpenCSVSerde'STORED AS INPUTFORMAT
 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat' OUTPUTFORMAT
 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat' LOCATION 'Copied S3 URI';
```

6. Pour effacer l'éditeur de requête, cliquez sur Clear (Effacer). Ensuite, chargez le rapport d'inventaire dans la table à l'aide de la commande suivante. Remplacez *table_name* par le nom que vous avez choisi lors de l'étape précédente. Choisissez ensuite Run (Exécuter) pour exécuter la requête.

```
MSCK REPAIR TABLE table_name;
```

7. Pour effacer l'éditeur de requête, cliquez sur Clear (Effacer). Exécutez la requête `SELECT` suivante pour récupérer toutes les entrées du rapport d'inventaire d'origine et remplacer tous les ID de version vides par des chaînes `null`. Remplacez *table_name* par le nom que vous avez choisi précédemment, et remplacez *YYYY-MM-DD-HH-MM* dans la clause `WHERE` par la date du rapport d'inventaire à laquelle vous souhaitez que cet outil soit exécuté. Choisissez ensuite Run (Exécuter) pour exécuter la requête.

```
SELECT bucket as Bucket, key as Key, CASE WHEN version_id = '' THEN 'null' ELSE
 version_id END as VersionId FROM table_name WHERE dt = 'YYYY-MM-DD-HH-MM';
```

8. Revenez à la console Amazon S3 (https://console.aws.amazon.com/s3/), et accédez au compartiment S3 que vous avez choisi précédemment pour Location of query result (Emplacement des résultats de la requête). Il devrait contenir une série de dossiers se terminant par la date.

   Par exemple, vous devriez voir quelque chose de semblable à s3://***DOC-EXAMPLE-BUCKET***/***query-result-location***/Unsaved/2021/10/07/. Vous devriez voir des fichiers `.csv` contenant les résultats de la requête `SELECT` que vous avez exécutée.

   Choisissez le fichier CSV avec la date de modification la plus récente. Téléchargez ce fichier sur votre ordinateur local pour la prochaine étape.

9. Le fichier CSV généré contient une ligne d'en-tête. Pour utiliser ce fichier CSV en tant qu'entrée pour une tâche S3 Batch Operations, vous devez supprimer la ligne d'en-tête, car Batch Operations ne prend pas en charge les lignes d'en-tête sur les manifestes CSV.

   Pour supprimer la ligne d'en-tête, vous pouvez exécuter l'une des commandes suivantes sur le fichier. Remplacez *`file.csv`* par le nom de votre fichier CSV.

   Pour les machines macOS et Linux, exécutez la commande `tail` dans une fenêtre Terminal.

   ```
   tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
   ```

   Pour les machines Windows, exécutez le script suivant dans une fenêtre Windows PowerShell. Remplacez *`File-location`* par le chemin de votre fichier et *`file.csv`* par le nom du fichier.

   ```
   $ins = New-Object System.IO.StreamReader File-location\file.csv
   $outs = New-Object System.IO.StreamWriter File-location\temp.csv
   try {
       $skip = 0
       while ( !$ins.EndOfStream ) {
           $line = $ins.ReadLine();
           if ( $skip -ne 0 ) {
               $outs.WriteLine($line);
           } else {
               $skip = 1
           }
   ```

```
    }
} finally {
    $outs.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. Après avoir supprimé la ligne d'en-tête du fichier CSV, vous êtes prêt à utiliser ce dernier comme manifeste dans une tâche S3 Batch Operations. Chargez le fichier CSV dans un compartiment S3 ou un emplacement de votre choix, puis créez une tâche Batch Operations en utilisant le fichier CSV comme manifeste.

    Pour de plus amples informations sur la création d'une tâche Batch Operations, veuillez consulter Création d'une tâche d'opérations par lot S3.

Utiliser le champ Liste ACL d'objet

Un rapport d'inventaire Amazon S3 contient une liste des objets figurant dans le compartiment source S3 et les métadonnées de chaque objet. Le champ Liste de contrôle d'accès (ACL) d'objet est un champ de métadonnées disponible dans l'inventaire Amazon S3. Plus précisément, le champ Liste ACL d'objet contient la liste de contrôle d'accès (ACL) de chaque objet. L'ACL d'un objet définit le Comptes AWS ou les groupes autorisés à accéder à cet objet et le type d'accès accordé. Pour plus d'informations, consultez Présentation de la liste de contrôle d'accès (ACL) et Liste d'inventaire Amazon S3.

Le champ Liste ACL de l'objet dans les rapports d'inventaire Amazon S3 est défini au format JSON. Les données JSON comprennent les champs suivants :

- `version` : version du format de champ Liste ACL de l'objet dans les rapports d'inventaire. Ce champ possède le format de date `yyyy-mm-dd`.
- `status` : les valeurs possibles sont `AVAILABLE` ou `UNAVAILABLE` pour indiquer si une liste ACL d'objet est disponible pour un objet. Lorsque le statut de la liste ACL de l'objet est `UNAVAILABLE`, la valeur du champ Propriétaire de l'objet dans le rapport d'inventaire est également `UNAVAILABLE`.
- `grants` : paires bénéficiaire-autorisation qui répertorient le statut d'autorisation de chaque bénéficiaire accordé par la liste ACL de l'objet. Les valeurs disponibles pour un bénéficiaire sont `CanonicalUser` et `Group`. Pour plus d'informations sur les bénéficiaires, consultez Bénéficiaires figurant dans les listes de contrôle d'accès.

Pour un bénéficiaire possédant le type `Group`, une paire bénéficiaire-autorisation inclut les attributs suivants :

- `uri` : un groupe Amazon S3 prédéfini.
- `permission` : les autorisations de liste ACL accordées sur l'objet. Pour plus d'informations, consultez Autorisations de liste ACL sur un objet.
- `type` : le type `Group`, qui indique que le bénéficiaire est un groupe.

Pour un bénéficiaire possédant le type `CanonicalUser`, une paire bénéficiaire-autorisation inclut les attributs suivants :

- `canonicalId` : une forme obfusquée de l'ID Compte AWS . L'ID utilisateur canonique d'un Compte AWS est spécifique à ce compte. Vous pouvez récupérer l'ID utilisateur canonique. Pour plus d'informations, voir Trouver l'identifiant d'utilisateur canonique correspondant à votre compte Compte AWS dans le Guide de référence de gestion de AWS compte.

  > **Note**
  >
  > Si le bénéficiaire d'une ACL est l'adresse e-mail d'un Compte AWS, S3 Inventory utilise cette adresse Compte AWS et le `CanonicalUser` type pour spécifier ce bénéficiaire. `canonicalId` Pour plus d'informations, consultez Bénéficiaires figurant dans les listes de contrôle d'accès.

- `permission` : les autorisations de liste ACL accordées sur l'objet. Pour plus d'informations, consultez Autorisations de liste ACL sur un objet.
- `type`— Le type`CanonicalUser`, qui indique que le bénéficiaire est un. Compte AWS

L'exemple suivant montre les valeurs possibles pour le champ Liste ACL d'objet au format JSON :

```
{
    "version": "2022-11-10",
    "status": "AVAILABLE",
    "grants": [{
        "uri": "http://acs.amazonaws.com/groups/global/AllUsers",
        "permission": "READ",
        "type": "Group"
    }, {
        "canonicalId": "example-canonical-id",
        "permission": "FULL_CONTROL",
```

```
        "type": "CanonicalUser"
    }]
}
```

> **Note**
>
> Le champ Liste ACL d'objet est défini au format JSON. Un rapport d'inventaire affiche la valeur du champ Liste ACL d'objet sous la forme d'une chaîne codée en base64. Supposons, par exemple, que vous disposiez du champ Liste ACL d'objet suivant au format JSON :
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```
>
> Le champ Liste ACL d'objet est codé et affiché sous la forme de la chaîne codée en base64 suivante :
>
> ```
> eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
> ```
>
> Pour obtenir la valeur décodée au format JSON pour le champ Liste ACL d'objet, vous pouvez interroger ce champ dans Amazon Athena. Pour d'autres exemples de requête, consultez Interrogation d'un inventaire Amazon S3 avec Amazon Athena.

Vue d'ensemble de la réplication d'objets

Vous pouvez utiliser la réplication pour activer la copie automatique et asynchrone d'objets dans des compartiments Amazon S3. Les compartiments configurés pour la réplication d'objet peuvent appartenir au même Compte AWS ou à des comptes distincts. Vous pouvez répliquer des objets vers un compartiment de destination unique ou plusieurs compartiments de destination. Les compartiments de destination peuvent se trouver dans une région différente Régions AWS ou dans la même région que le compartiment source.

Il existe deux types de réplication : la réplication en direct et la réplication à la demande.

- Réplication en direct : pour répliquer automatiquement les objets nouveaux et mis à jour au fur et à mesure qu'ils sont écrits dans le compartiment source, utilisez la réplication en direct. La réplication dynamique ne réplique aucun objet qui existait dans le compartiment avant que vous ne configuriez la réplication. Pour répliquer des objets qui existaient avant de configurer la réplication, utilisez la réplication à la demande.
- Réplication à la demande : pour répliquer des objets existants depuis le compartiment source vers un ou plusieurs compartiments de destination à la demande, utilisez S3 Batch Replication. Pour en savoir plus sur la réplication d'objets existants, consultez Quand utiliser la réplication par lot S3.

Il existe deux formes de réplication dynamique : la réplication entre régions (CRR) et la réplication entre régions (SRR).

- Réplication entre régions (CRR) : vous pouvez utiliser la CRR pour répliquer des objets dans différents compartiments Amazon S3. Régions AWS Pour plus d'informations sur le CRR, consultezthe section called “Quand utiliser la réplication entre Régions”.
- Réplication par région unique (SRR) : vous pouvez utiliser SRR pour copier des objets dans des compartiments Amazon S3 au même endroit. Région AWS Pour plus d'informations sur le SRR, consultezthe section called “Quand utiliser la réplication au sein d'une même Région”.

Rubriques

- Réplication d'objets existants via la réplication par lot S3

Pourquoi utiliser la réplication ?

La réplication entre Régions peut vous aider à réaliser les tâches suivantes :

- Répliquer des objets tout en conservant les métadonnées – Vous pouvez utiliser la réplication pour faire des copies de vos objets qui conservent toutes les métadonnées, telles que les heures de création des objets d'origine et les ID de version. Cette fonctionnalité est importante si vous devez vous assurer que votre réplica est identique à l'objet source.
- Répliquer des objets dans différentes classes de stockage – Vous pouvez utiliser la réplication pour placer directement des objets dans S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou une autre classe dans les compartiments de destination. Vous pouvez également répliquer vos données dans la même classe de stockage et utiliser des configurations de cycle de vie sur les compartiments de destination pour déplacer vos objets vers une classe de stockage à froid si leur ancienneté le justifie.
- Conservez des copies d'objets appartenant à des propriétaires différents : quel que soit le propriétaire de l'objet source, vous pouvez demander à Amazon S3 de transférer le propriétaire de la Compte AWS réplique au propriétaire du compartiment de destination. Il s'agit de l'option de substitution du propriétaire Vous pouvez utiliser cette option pour limiter l'accès aux réplicas d'objets.
- Stockez les objets sur plusieurs sites Régions AWS : pour garantir les différences géographiques quant à l'emplacement de conservation de vos données, vous pouvez définir plusieurs compartiments de destination dans différents Régions AWS domaines. Cette fonctionnalité peut vous aider à répondre à certaines exigences de conformité.
- Répliquer des objets en 15 minutes : pour répliquer vos données dans la même région Région AWS ou entre différentes régions dans un délai prévisible, vous pouvez utiliser S3 Replication Time Control (S3 RTC). Le contrôle du délai de réplication S3 permet de répliquer 99,99 % des nouveaux objets stockés dans Simple Storage Service (Amazon S3) dans les 15 minutes (conformément à un contrat de niveau de service (SLA)). Pour plus d'informations, consultez the section called “Utiliser le contrôle du délai de réplication S3”.

> **Note**
>
> Le contrôle du délai de réplication S3 ne s'applique pas à la réplication par lot. La réplication par lot est une tâche de réplication à la demande et peut être suivie avec les

opérations par lot S3. Pour plus d'informations, consultez Suivi de l'état de la tâche et des rapports de fin de tâche.

- Synchroniser des compartiments, répliquer des objets existants et répliquer des objets précédemment défaillants ou répliqués – Pour synchroniser des compartiments et répliquer des objets existants, utilisez la réplication par lot comme action de réplication à la demande. Pour plus d'informations sur les conditions d'utilisation de la réplication par lot, consultez Quand utiliser la réplication par lot S3.
- Répliquer des objets et basculer vers un compartiment dans une autre Région AWS : pour que toutes les métadonnées et tous les objets restent synchronisés entre les compartiments pendant la réplication des données, utilisez des règles de réplication bidirectionnelle avant de configurer les contrôles de basculement du point d'accès multi-régions d'Amazon S3. Les règles de réplication bidirectionnelle permettent de s'assurer que lorsque des données sont écrites dans le compartiment S3 vers lequel le trafic bascule, ces données sont ensuite répliquées vers le compartiment source.

Quand utiliser la réplication entre Régions

La réplication entre Régions (CRR) S3 permet de copier des objets entre des compartiments Amazon S3 situés dans différentes Régions AWS. La réplication entre Régions aide à :

- Respecter les exigences de conformité – Bien que Simple Storage Service (Amazon S3) stocke vos données dans plusieurs zones de disponibilité distantes géographiquement par défaut, les exigences de conformité peuvent vous obliger à stocker les données séparées par des distances encore plus importantes. Pour satisfaire à ces exigences, la réplication entre Régions vous permet de répliquer des données entre des Régions AWS distantes.
- Minimiser le temps de latence — Si vos clients se trouvent dans deux zones géographiques, vous pouvez minimiser le temps de latence lors de l'accès aux objets en Régions AWS conservant des copies d'objets géographiquement plus proches de vos utilisateurs.
- Améliorez l'efficacité opérationnelle : si vous avez des clusters de calcul répartis dans deux régions différentes Régions AWS qui analysent le même ensemble d'objets, vous pouvez choisir de conserver des copies d'objets dans ces régions.

Quand utiliser la réplication au sein d'une même Région

La réplication dans une même Région (SRR) est utilisée pour copier des objets entre compartiments Amazon S3 d'une même Région AWS. La SRR vous aide à :

- Regrouper les journaux dans un même compartiment – Si vous stockez des journaux dans plusieurs compartiments ou sur différents comptes, vous pouvez facilement répliquer les journaux dans un même compartiment d'une Région. Cela permet de simplifier le traitement des journaux dans un seul emplacement.
- Configurer la réplication en direct entre les comptes de production et de test – Si vous, ou vos clients, disposez de comptes de production et de test qui utilisent les mêmes données, vous pouvez répliquer des objets entre différents comptes, tout en conservant les métadonnées d'objet.
- Respectez les lois sur la souveraineté des données — Vous pourriez être amené à stocker plusieurs copies de vos données séparément Comptes AWS dans une région donnée. La réplication dans une même Région peut vous aider à répliquer des données critiques lorsque les exigences de conformité n'autorisent pas que les données sortent de votre pays.

Quand utiliser la réplication à double sens (réplication bi-directionnelle)

- Créez des ensembles de données partagés sur plusieurs Régions AWS : grâce à la synchronisation des modifications des répliques, vous pouvez facilement répliquer les modifications de métadonnées, telles que les listes de contrôle d'accès aux objets (ACL), les balises d'objets ou les verrous d'objets, sur les objets de réplication. Cette réplication bidirectionnelle est importante si vous voulez que tous les objets et les modifications des métadonnées des objets soient synchronisés. Vous pouvez activer la synchronisation de modification de réplica sur une règle de réplication nouvelle ou existante lors de l'exécution d'une réplication bidirectionnelle entre deux ou plusieurs compartiments dans des Régions AWS identiques ou différentes.
- Maintenez les données synchronisées entre les régions pendant le basculement : vous pouvez synchroniser les données dans des compartiments entre elles en Régions AWS configurant des règles de réplication bidirectionnelle avec la réplication inter-régions (CRR) S3 directement à partir d'un point d'accès multirégional. Pour prendre une décision éclairée quant au moment de lancer le basculement, vous pouvez également activer les métriques de réplication S3 afin de pouvoir

surveiller la réplication dans Amazon CloudWatch, dans S3 Replication Time Control (S3 RTC) ou depuis le point d'accès multirégional.

- Make your application highly available (Rendre votre application hautement disponible) : même en cas de perturbation du trafic régional, vous pouvez utiliser des règles de réplication bidirectionnelle pour que toutes les métadonnées et tous les objets restent synchronisés entre les compartiments pendant la réplication des données.

Quand utiliser la réplication par lot S3

La réplication par lot réplique des objets existants dans différents compartiments (option à la demande). Contrairement à la réplication en direct, ces tâches peuvent être exécutées selon les besoins. La réplication par lot peut vous aider à réaliser les tâches suivantes :

- Répliquer des objets existants – Vous pouvez utiliser la réplication par lot pour répliquer des objets qui ont été ajoutés au compartiment avant la configuration de la réplication dans une même région ou de la réplication entre Régions.
- Répliquer les objets qui n'ont pas pu être répliqués auparavant – Vous pouvez filtrer une tâche de réplication par lot pour tenter de répliquer des objets dont le statut de réplication a ÉCHOUÉ.
- Répliquer des objets déjà répliqués – Vous pourriez devoir stocker plusieurs copies de vos données dans des Comptes AWS ou des Régions AWS distincts. La réplication par lot peut répliquer des objets existants vers des destinations nouvellement ajoutées.
- Répliquer des réplicas d'objets créés à partir d'une règle de réplication – Les configurations de réplication créent des réplicas d'objets dans des compartiments de destination. Les réplicas d'objets peuvent être répliqués uniquement via la réplication par lot.

Exigences relatives à la charge de travail et réplication en direct

Selon les exigences de votre charge de travail, certains types de réplication seront mieux adaptés à votre cas d'utilisation que d'autres. Utilisez le tableau suivant pour déterminer le type de réplication à utiliser en fonction de votre situation et pour déterminer s'il convient d'utiliser S3 Replication Time Control (S3 RTC) pour votre charge de travail. S3 RTC réplique 99,99 % des nouveaux objets stockés dans Amazon S3 en 15 minutes (dans le cadre d'un accord de niveau de service, ou SLA). Pour plus d’informations, consultez the section called “Utiliser le contrôle du délai de réplication S3”.

Exigences de charge de travail pour la comparaison des réplications

Charge de travail requise	S3 RTC (SLA de 15 minutes)	Réplication entre régions (CRR)	Réplication à région unique (SRR)
Répliquer des objets entre différents Comptes AWS	Oui	Oui	Oui
Répliquez des objets au même endroit dans Région AWS un délai de 24 à 48 heures (sans respecter les SLA)	Non	Non	Oui
Répliquez des objets entre différents Régions AWS dans un délai de 24 à 48 heures (non garanti par un SLA)	Non	Oui	Non
Temps de réplication prévisible : soutenu par le SLA pour répliquer 99,9 % des objets en 15 minutes	Oui	Non	Non

Ce qui est répliqué par Amazon S3

Dans les compartiments, Simple Storage Service (Amazon S3) réplique uniquement les éléments spécifiques qui sont configurés pour la réplication.

Rubriques

- Qu'est-ce qui est répliqué avec les configurations de réplication ?
- Qu'est-ce qui n'est pas répliqué avec les configurations de réplication ?

- Comment le chiffrement par défaut du compartiment a un impact sur la réplication

Qu'est-ce qui est répliqué avec les configurations de réplication ?

Par défaut, Simple Storage Service (Amazon S3) réplique les éléments suivants :

- Objets créés après l'ajout d'une configuration de réplication.
- Objets non chiffrés.
- Objets chiffrés à l'aide de clés fournies par le client (SSE-C), objets chiffrés au repos sous une clé gérée Amazon S3 (SSE-S3) ou une clé KMS stockée dans AWS Key Management Service (SSE-KMS). Pour plus d'informations, consultez the section called "Réplication d'objets chiffrés".
- Métadonnées d'objet des objets source vers les réplicas. Pour plus d'informations sur la réplique des métadonnées des réplicas vers les objets source, consultez la section Répliquer les modifications de métadonnées avec la synchronisation des modifications de réplica Amazon S3.
- Seuls les objets du compartiment source pour lesquels le propriétaire du compartiment dispose d'autorisations en lecture sur ces objets et leurs listes ACL.

  Pour de plus informations sur la propriété des ressources, veuillez consulter Propriété du compartiment et de l'objet Amazon S3.
- Les mises à jour des listes ACL d'objet, sauf si vous indiquez à Amazon S3 de modifier le propriétaire des réplicas lorsque les compartiments source et de destination n'appartiennent pas aux mêmes comptes.

  Pour plus d'informations, consultez Modification du propriétaire d'un réplica.

  Simple Storage Service (Amazon S3) peut mettre un certain temps à synchroniser les deux listes ACL. Ce changement de propriété s'applique uniquement aux objets créés après l'ajout d'une configuration de réplication dans le compartiment.
- Les balises d'objets, le cas échéant.
- Les informations de conservation du verrouillage des objets S3, le cas échéant.

  Quand Amazon S3 réplique des objets pour lesquels des informations de conservation ont été appliquées, il applique ces mêmes contrôles de conservation à vos réplicas, remplaçant ainsi la période de conservation par défaut configurée sur vos compartiments de destination. Si vous n'avez pas de contrôles de conservation appliqués aux objets de votre compartiment source, et que vous répliquez dans un compartiment de destination dont la période de conservation par

défaut est définie, la période de conservation par défaut du compartiment de destination est appliquée à vos répliques d'objets. Pour plus d'informations, consultez Utilisation du verrouillage des objets S3.

Impact des opérations de suppression sur la réplication

Si vous supprimez un objet du compartiment source, les actions suivantes se produisent par défaut :

- Si vous émettez une demande de suppression (DELETE) sans spécifier d'ID de version d'objet, Amazon S3 ajoute un marqueur de suppression. Amazon S3 traite le marqueur de suppression comme suit :
  - Si vous utilisez la dernière version de la configuration de réplication, en spécifiant l'élément `Filter` dans une règle de configuration de réplication, Simple Storage Service (Amazon S3) ne réplique pas le marqueur de suppression par défaut. Vous pouvez toutefois ajouter la réplication des marqueurs de suppression aux non-tag-based règles. Pour plus d'informations, consultez Répliquer des marqueurs de suppression entre les compartiments.
  - Si vous ne spécifiez pas l'élément `Filter`, Simple Storage Service (Amazon S3) suppose que la configuration de réplication est la version V1 et réplique les marqueurs de suppression résultant des actions de l'utilisateur. Toutefois, si Amazon S3 supprime un objet en raison d'une action de cycle de vie, le marqueur de suppression n'est pas répliqué dans les compartiments de destination.
- Si vous spécifiez un ID de version d'objet à supprimer dans une requête `DELETE`, Amazon S3 supprime cette version de l'objet dans le compartiment source. Mais le service ne réplique pas la suppression dans les compartiments de destination. En d'autres termes, il ne supprime pas la même version de l'objet dans les compartiments de destination. Les données sont ainsi protégées contre les suppressions malveillantes.

Qu'est-ce qui n'est pas répliqué avec les configurations de réplication ?

Par défaut, Simple Storage Service (Amazon S3) ne réplique pas les éléments suivants :

- Les objets du compartiment source qui sont des réplicas ayant été créés par une autre règle de réplication. Supposons, par exemple, que vous configurez une réplication où le compartiment A est le compartiment source et le compartiment B celui de destination. Supposons ensuite que vous ajoutez une autre configuration de réplication où le compartiment B est le compartiment source et le compartiment C celui de destination. Dans ce cas, les objets du compartiment B qui sont les réplicas d'objets du compartiment A ne sont pas répliqués dans le compartiment C.

Pour répliquer des objets qui sont des réplicas, utilisez la réplication par lot. Pour en savoir plus sur la configuration de la réplication par lot, consultez Réplication d'objets existants.

- Objets du compartiment source qui ont déjà été répliqués vers une autre destination. Par exemple, si vous changez le compartiment de destination dans une configuration de réplication existante, Simple Storage Service (Amazon S3) ne procède pas à une nouvelle réplication des objets.

  Pour répliquer des objets précédemment répliqués, utilisez la réplication par lot. Pour en savoir plus sur la configuration de la réplication par lot, consultez Réplication d'objets existants.

- La réplication par lot ne prend pas en charge la réplication répétée d'objets qui ont été supprimés avec l'ID de version de l'objet dans le compartiment de destination. Pour répéter la réplication de ces objets, vous pouvez copier les objets sources en place avec une tâche de copie par lot. La copie de ces objets en place crée de nouvelles versions des objets dans le compartiment source et lance automatiquement la réplication vers la destination. Pour plus d'informations sur l'utilisation de Batch Copy, veuillez consulter Exemples qui utilisent des opérations par lot pour copier des objets.

- Par défaut, lors de la réplication à partir d'un autre compartiment Compte AWS, les marqueurs de suppression ajoutés au compartiment source ne sont pas répliqués.

  Pour savoir comment répliquer des marqueurs de suppression, consultez Répliquer des marqueurs de suppression entre les compartiments.

- Objets stockés dans les classes ou niveaux de stockage S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive Access, S3 Intelligent-Tiering Archive Access ou S3 Intelligent-Tiering Deep Archive Access. Vous ne pouvez pas répliquer ces objets tant que vous ne les avez pas restaurés et copiés dans une autre classe de stockage.

  Pour en savoir plus sur S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive, consultezClasses de stockage pour les objets rarement consultés.

  Pour en savoir plus sur la hiérarchisation intelligente S3, consultez. Amazon S3 Intelligent Tiering

- Objets dans le compartiment source pour lesquels le propriétaire du compartiment ne dispose pas des autorisations de réplication suffisantes.

  Pour obtenir des informations sur la façon dont un propriétaire d'objet peut accorder des autorisations à un propriétaire de compartiment, consultez Octroi d'autorisations intercomptes pour charger des objets tout en garantissant que le propriétaire du compartiment dispose d'un contrôle total.

- Les mises à jour des sous-ressources de niveau compartiment.

Par exemple, si vous modifiez la configuration du cycle de vie ou ajoutez une configuration de notification à votre compartiment source, ces modifications ne sont pas appliquées au compartiment de destination. Cette fonction permet ainsi d'avoir des configurations différentes dans les compartiments source et de destination.

- Actions effectuées par la configuration du cycle de vie.

  Par exemple, si la configuration du cycle de vie est activée uniquement sur votre compartiment source, Simple Storage Service (Amazon S3) crée des marqueurs de suppression pour les objets expirés, mais ne réplique pas ces marqueurs. Si vous souhaitez appliquer la même configuration de cycle de vie aux compartiments source et de destination, activez la même configuration de cycle de vie sur les deux. Pour en savoir plus sur la configuration du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

- Lorsque vous utilisez des règles de réplication basées sur des balises avec une réplication en direct, les nouveaux objets doivent être étiquetés avec la balise de règle de réplication correspondante lors de l'`PutObject`opération. Dans le cas contraire, les objets ne seront pas répliqués. Si des objets sont étiquetés après l'`PutObject`opération, ils ne seront pas non plus répliqués.

  Pour répliquer des objets marqués après l'`PutObject`opération, vous devez utiliser S3 Batch Replication. Pour en savoir plus sur la réplication par lot, consultez Réplication d'objets existants.

Comment le chiffrement par défaut du compartiment a un impact sur la réplication

Après avoir activé le chiffrement par défaut pour un compartiment de destination de réplication, le comportement de chiffrement suivant s'applique :

- Si des objets du compartiment source ne sont pas chiffrés, les objets réplica du compartiment de destination sont chiffrés à l'aide des paramètres de chiffrement par défaut du compartiment de destination. Par conséquent, les balises d'entité (ETags) des objets sources diffèrent des ETags des objets réplica. Si certaines de vos applications utilisent des ETags, vous devez les mettre à jour pour tenir compte de cette différence.
- Si les objets du compartiment source sont chiffrés à l'aide d'un chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3), d'un chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou d'un chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS), les objets répliques du compartiment de destination utilisent

le même type de chiffrement que les objets source. Les paramètres de chiffrement par défaut du compartiment de destination ne sont pas utilisés.

Exigences et considérations relatives à la réplication

La réplication Amazon S3 nécessite les éléments suivants :

- Le propriétaire du compartiment source doit avoir Régions AWS activé la source et la destination pour son compte. Le propriétaire du compartiment de destination doit avoir activé la Région de destination pour son compte.

  Pour plus d'informations sur l'activation ou la désactivation d'un Région AWS, voir Gestion Régions AWS dans le Références générales AWS.
- La gestion des versions doit être activée pour les compartiments source et de destination. Pour plus d'informations sur la gestion des versions, consultez Utilisation de la gestion des versions dans les compartiments S3.
- Simple Storage Service (Amazon S3) doit disposer des autorisations adéquates pour répliquer en votre nom les objets issus du compartiment source vers le ou les compartiments de destination. Pour plus d'informations sur ces autorisations, consultez Configuration des autorisations pour la réplication en direct.
- Si le propriétaire du compartiment source ne possède pas l'objet dans le compartiment, le propriétaire de l'objet doit accorder au propriétaire du compartiment les autorisations `READ` et `READ_ACP` avec la liste de contrôle d'accès (ACL) de l'objet. Pour plus d'informations, consultez Présentation de la liste de contrôle d'accès (ACL).
- Si le verrouillage des objets S3 est activé dans le compartiment source, alors il doit également l'être dans les compartiments de destination.

  Pour activer la réplication sur un compartiment sur lequel Object Lock est activé, vous devez utiliser l' AWS Command Line Interface API REST ou AWS les SDK. Pour plus d'informations générales, consultez Utilisation du verrouillage des objets S3.

  > **Note**
  >
  > Vous devez accorder deux nouvelles autorisations sur le compartiment S3 source dans le rôle AWS Identity and Access Management (IAM) que vous utilisez pour configurer la réplication. Les deux nouvelles autorisations sont `s3:GetObjectRetention` et `s3:GetObjectLegalHold`. Si le rôle dispose d'une autorisation `s3:Get*`, il répond aux

exigences. Pour plus d'informations, consultez Configuration des autorisations pour la réplication en direct.

Pour plus d'informations, consultez Configuration de la réplication en direct.

Si vous définissez la configuration de réplication dans un scénario à plusieurs comptes, où les compartiments source et de destination appartiennent à différents Comptes AWS, la condition supplémentaire suivante s'applique :

- Le propriétaire des compartiments de destination doit accorder des autorisations au propriétaire du compartiment source lui permettant de répliquer des objets avec une stratégie de compartiment. Pour plus d'informations, consultez Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS.
- Les compartiments de destination ne peuvent pas être configurés en tant que compartiment Paiement par le demandeur. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.

Considérations relatives à la réplication

Avant de créer une configuration de réplication, tenez compte des points suivants.

Rubriques

- Configuration de cycle de vie et réplicas d'objets
- Configuration de la gestion des versions et configuration de la réplication
- Utiliser la réplication S3 avec S3 Intelligent-Tiering
- Configuration de la journalisation et configuration de la réplication
- Réplication entre Régions et Région de destination
- Réplication par lots S3
- Contrôle du temps de réplication S3

Configuration de cycle de vie et réplicas d'objets

Le temps nécessaire à Amazon S3 pour répliquer un objet dépend de la taille de ce dernier. Pour les objets volumineux, cela peut prendre plusieurs heures, Même si la mise à disposition d'un réplica

dans la destination peut prendre du temps, la création du réplica prend autant de temps que la création de l'objet correspondant dans le compartiment source. Si une configuration de cycle de vie est activée dans le compartiment de destination, les règles de cycle de vie tiennent compte du moment de création de l'objet à l'origine, et non pas du moment où le réplica est devenu disponible dans le compartiment de destination.

La configuration de réplication nécessite que le compartiment soit activé pour la gestion des versions. Lorsque vous activez la gestion des versions pour un compartiment, considérez les éléments suivants :

- Si vous avez une configuration de cycle de vie d'expiration des objets, après avoir activé la gestion des versions, ajoutez une stratégie `NonCurrentVersionExpiration` pour conserver le même comportement de suppression définitive que celui défini avant l'activation de la gestion des versions.
- Si vous avez une configuration de cycle de vie de transition, après avoir activé la gestion des versions, envisagez d'ajouter une politique `NonCurrentVersionTransition`.

Configuration de la gestion des versions et configuration de la réplication

La gestion des versions doit être activée pour les deux compartiments source et de destination lorsque vous configurez une réplication dans un compartiment. Après avoir activé la gestion des versions dans les compartiments source et de destination, et configuré la réplication dans le compartiment source, les problèmes suivants se produisent :

- Si vous tentez de désactiver la gestion des versions sur le compartiment source, Amazon S3 renvoie une erreur. Vous devez supprimer la configuration de réplication avant de désactiver la gestion des versions dans le compartiment source.
- Si vous désactivez la gestion des versions dans le compartiment de destination, la réplication échoue. L'objet source a le statut de réplication `FAILED`.

Utiliser la réplication S3 avec S3 Intelligent-Tiering

La classe de stockage S3 Intelligent-Tiering est conçue pour optimiser les coûts de stockage en transférant automatiquement les données vers le niveau d'accès le plus économique. Moyennant des frais mensuels minimes de surveillance et d'automatisation des objets, S3 Intelligent-Tiering surveille les modèles d'accès et déplace automatiquement les objets qui n'ont pas été consultés vers des niveaux d'accès moins coûteux.

La réplication d'objets stockés dans S3 Intelligent-Tiering avec la réplication par lot S3 ou l'invocation de CopyObject ou de UploadPartCopy constitue un accès. Dans ces cas, les objets source des opérations de copie ou de réplication sont hiérarchisés.

Pour plus d'informations sur S3 Intelligent-Tiering, consultez Amazon S3 Intelligent Tiering.

Configuration de la journalisation et configuration de la réplication

Si Amazon S3 livre des journaux à un compartiment où la réplication est activée, il réplique les objets journaux.

Si les journaux d'accès au serveur (Enregistrement de demandes avec journalisation des accès au serveur) ou les journaux AWS CloudTrail (Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail) sont activés sur votre compartiment source ou de destination, Amazon S3 inclut les demandes liées à la réplication dans les journaux. Par exemple, Amazon S3 consigne chaque objet qu'il réplique.

Réplication entre Régions et Région de destination

Amazon S3 Cross-Region Replication (CRR) est utilisé pour copier des objets dans différents compartiments S3. Régions AWS Vous pouvez choisir la Région de votre compartiment de destination en fonction de vos besoins professionnels ou de considérations tarifaires. Par exemple, les frais de transfert de données entre Régions varient en fonction des Régions que vous choisissez.

Supposons que vous avez choisi USA Est (Virginie du Nord) (us-east-1) comme Région de votre compartiment source. Si vous choisissez USA Ouest (Oregon) (us-west-2) comme Région pour votre compartiment de destination, vous payez plus que si vous choisissez la Région USA Est (Ohio) (us-east-2). Pour obtenir des informations sur la tarification, veuillez consulter « Tarification du transfert de données » dans Tarification Amazon S3.

Aucun frais de transfert de données n'est associé à la réplication dans une même Région (SRR).

Réplication par lots S3

Pour plus d'informations sur les considérations relatives à la réplication par lots, consultezConsidérations sur la réplication par lot S3.

Contrôle du temps de réplication S3

Pour plus d'informations sur les meilleures pratiques et les considérations relatives au contrôle du temps de réplication S3 (S3 RTC), consultezBonnes pratiques et directives de contrôle du délai de réplication S3.

Configuration de la réplication en direct

> **Note**
>
> Les objets qui existaient avant la configuration de la réplication ne sont pas répliqués automatiquement. En d'autres termes, Simple Storage Service (Amazon S3) ne réplique pas les objets de manière rétroactive. Pour répliquer des objets créés avant la configuration de la réplication, utilisez la réplication par lot S3. Pour en savoir plus sur la configuration de la réplication par lot, consultez Réplication d'objets existants.

Pour activer la réplication en direct (réplication dans la même région (SRR) ou réplication entre régions (CRR), ajoutez une configuration de réplication à votre compartiment source. Cette configuration indique à Amazon S3 de répliquer les objets comme indiqué. Dans la configuration de réplication, vous devez renseigner les éléments suivants :

- Compartiments de destination – Compartiments dans lesquels vous souhaitez qu'Simple Storage Service (Amazon S3) réplique les objets.
- Objets que vous voulez répliquer – Vous pouvez répliquer l'ensemble des objets du compartiment source ou un sous-ensemble. Vous identifiez un sous-ensemble en fournissant un préfixe de nom de clé, une ou plusieurs balises d'objets, ou les deux dans la configuration.

  Par exemple, si vous configurez une règle de réplication pour ne répliquer que les objets dotés du préfixe de nom de clé `Tax/`, Amazon S3 réplique les objets avec les clés `Tax/doc1` et `Tax/doc2`. Mais le service ne réplique pas les objets dotés d'une clé `Legal/doc3`. Si vous spécifiez un préfixe et une ou plusieurs balises, Simple Storage Service (Amazon S3) réplique uniquement les objets dotés du préfixe de clé et des balises spécifiques.
- Un rôle AWS Identity and Access Management (IAM) : Amazon S3 assume ce rôle IAM pour répliquer des objets en votre nom.

Outre ces conditions minimales requises, vous pouvez choisir les options suivantes :

- Classe de stockage des réplicas – Par défaut, Simple Storage Service (Amazon S3) stocke les réplicas d'objet dans la même classe de stockage que celle de l'objet source. Vous pouvez spécifier une classe de stockage différente pour les réplicas.
- Propriété des réplicas – Simple Storage Service (Amazon S3) suppose qu'un réplica d'objet appartient toujours au propriétaire de l'objet source. Ainsi, lorsqu'il réplique des objets, il réplique

également la liste de contrôle d'accès (ACL) correspondante ou le paramètre de propriété de l'objet S3. Si les compartiments source et de destination appartiennent à des Comptes AWS différents, vous pouvez configurer la réplication de sorte à remplacer le propriétaire d'un réplica par le Compte AWS qui possède le compartiment de destination.

Vous pouvez configurer la réplication à l'aide de l'API REST, AWS des SDK AWS Command Line Interface (AWS CLI) ou de la console Amazon S3.

Simple Storage Service (Amazon S3) fournit également des opérations d'API pour prendre en charge la configuration des règles de réplication. Pour plus d'informations, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service :

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Rubriques

Configuration de réplication

Amazon S3 stocke une configuration de réplication au format XML. Dans le fichier XML de configuration de réplication, vous spécifiez un rôle AWS Identity and Access Management (IAM) et une ou plusieurs règles.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

Amazon S3 ne peut pas répliquer d'objets sans votre autorisation. Vous accordez des autorisations avec le rôle IAM que vous spécifiez dans la configuration de réplication. Amazon S3 endosse le rôle IAM pour répliquer des objets en votre nom. Vous devez accorder les autorisations requises au rôle IAM en premier. Pour plus d'informations sur la gestion des autorisations, consultez Configuration des autorisations pour la réplication en direct.

Vous ajoutez une règle dans la configuration de réplication pour les scénarios suivants :

- Vous souhaitez répliquer tous les objets.
- Vous souhaitez répliquer un sous-ensemble d'objets. Vous identifiez le sous-ensemble d'objets en ajoutant un filtre dans la règle. Dans le filtre, vous spécifiez un préfixe de clé d'objet et/ou des balises pour identifier le sous-ensemble d'objets auquel la règle s'applique. Les filtres ciblent les objets qui correspondent exactement aux valeurs que vous avez spécifiées.

Vous ajoutez plusieurs règles dans une configuration de réplication si vous souhaitez répliquer un sous-ensemble d'objets distinct. Dans chaque règle, vous spécifiez un filtre qui sélectionne un sous-ensemble d'objets différent. Par exemple, vous pouvez choisir de répliquer des objets qui possèdent les préfixes de clé `tax/` ou `document/`. Pour ce faire, vous ajoutez deux règles, l'une qui spécifie le filtre de préfixe de clé `tax/` et l'autre qui spécifie le préfixe de clé `document/`. Pour en savoir plus sur le préfixe de clé d'objet, consultez Organisation des objets à l'aide de préfixes.

Les sections suivantes fournissent des informations supplémentaires.

Rubriques

Configuration de base d'une règle

Chaque règle doit inclure le statut et la priorité de la règle. La règle doit également indiquer s'il faut répliquer les marqueurs de suppression.

- `Status` indique si la règle est activée ou désactivée via les valeurs `Enabled` ou `Disabled`. Si une règle est désactivée, Simple Storage Service (Amazon S3) n'effectue pas les actions spécifiées dans la règle.
- `Priority` indique quelle règle a priorité en cas de conflit de deux règles de réplication ou plus. Simple Storage Service (Amazon S3) tente de répliquer les objets selon toutes les règles de réplication. Toutefois, s'il existe deux règles ou plus avec le même compartiment de destination, les objets sont répliqués selon la règle avec la priorité la plus élevée. Plus le nombre est élevé, plus la priorité est haute.
- `DeleteMarkerReplication` indique s'il faut répliquer les marqueurs de suppression via les valeurs `Enabled` ou `Disabled`.

Dans la configuration de destination, vous devez fournir le nom du compartiment ou des compartiments dans lesquels vous voulez qu'Simple Storage Service (Amazon S3) réplique les objets.

L'exemple suivant indique les conditions minimales requises pour une règle V2. Pour assurer la compatibilité descendante, Amazon S3 continue de prendre en charge le format XML V1. Pour plus d'informations, consultez Rétrocompatibilité.

```
...
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled-or-Disabled</Status>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Priority>integer</Priority>
        <DeleteMarkerReplication>
           <Status>Enabled-or-Disabled</Status>
        </DeleteMarkerReplication>
        <Destination>
           <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
        </Destination>
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
...
```

Vous pouvez également spécifier d'autres options de configuration. Par exemple, vous pouvez choisir d'utiliser une classe de stockage pour les réplicas d'objets qui diffèrent de la classe associée à l'objet source.

Facultatif : Spécification d'un filtre

Pour choisir un sous-ensemble d'objets auquel la règle s'applique, ajoutez un filtre facultatif. Vous pouvez filtrer par préfixe de clé d'objet et/ou par balises d'objets. Si vous filtrez par préfixe de clé et par balises d'objet, Amazon S3 combine ces filtres au moyen de l'opérateur logique `AND`. En d'autres termes, la règle s'applique à un sous-ensemble d'objets doté d'un préfixe de clé et de balises spécifiques.

Filtre basé sur un préfixe de clé d'objet

Pour spécifier une règle dont le filtre est défini sur un préfixe de clé d'objet, utilisez le code suivant. Vous ne pouvez spécifier qu'un seul préfixe.

```
<Rule>
    ...
    <Filter>
        <Prefix>key-prefix</Prefix>
    </Filter>
    ...
</Rule>
...
```

Filtre basé sur des balises d'objet

Pour spécifier une règle dont le filtre est défini sur des balises d'objets, utilisez le code suivant. Vous pouvez spécifier une ou plusieurs balises d'objets.

```
<Rule>
    ...
    <Filter>
        <And>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
```

```
            </Tag>
             ...
        </And>
    </Filter>
    ...
</Rule>
...
```

Filtre basé sur un préfixe de clé et des balises d'objet

Pour spécifier un filtre de règle défini sur un préfixe de clé et des balises d'objets, utilisez le code suivant. Vous enveloppez ces filtres dans un élément parent <And>. Amazon S3 effectue une opération logique AND pour combiner ces filtres. En d'autres termes, la règle s'applique à un sous-ensemble d'objets doté à la fois d'un préfixe de clé et de balises spécifiques.

```
<Rule>
    ...
    <Filter>
        <And>
            <Prefix>key-prefix</Prefix>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
    </Filter>
    ...
</Rule>
...
```

> **Note**
>
> - Si vous spécifiez une règle avec un <Filter> élément vide, celle-ci s'applique à tous les objets de votre compartiment.
> - Lorsque vous utilisez des règles de réplication basées sur des balises avec une réplication en direct, les nouveaux objets doivent être étiquetés avec la balise de règle de réplication correspondante lors de l'PutObjectopération. Dans le cas contraire, les objets ne seront

pas répliqués. Si des objets sont étiquetés après l'`PutObject`opération, ils ne seront pas non plus répliqués.

Pour répliquer des objets marqués après l'`PutObject`opération, vous devez utiliser S3 Batch Replication. Pour en savoir plus sur la réplication par lot, consultez Réplication d'objets existants.

Configurations de destinations supplémentaires

Dans la configuration de destination, vous spécifiez le ou les compartiments dans lesquels vous voulez qu'Amazon S3 réplique les objets. Vous pouvez définir des configurations pour répliquer des objets d'un compartiment source dans un ou plusieurs compartiments de destination.

```
...
<Destination>
    <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
</Destination>
...
```

Vous pouvez ajouter les options suivantes dans l'élément `<Destination>` :

Rubriques

- Spécifier une classe de stockage
- Ajouter plusieurs compartiments de destination
- Spécifier des paramètres différents pour chaque règle de réplication avec plusieurs compartiments de destination
- Modification du propriétaire d'un réplica
- Activer le contrôle du délai de réplication S3
- Répliquez les objets créés avec le chiffrement côté serveur en utilisant AWS KMS

Spécifier une classe de stockage

Vous pouvez spécifier la classe de stockage pour les réplicas d'objets. Par défaut, Amazon S3 utilise la classe de stockage de l'objet source pour créer les réplicas d'objets, comme dans l'exemple ci-dessous.

```
...
```

```
<Destination>
       <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Ajouter plusieurs compartiments de destination

Vous pouvez ajouter plusieurs compartiments de destination dans une configuration de réplication unique, comme suit.

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
    </Destination>
</Rule>
...
```

Spécifier des paramètres différents pour chaque règle de réplication avec plusieurs compartiments de destination

Lorsque vous ajoutez plusieurs compartiments de destination dans une configuration de réplication unique, vous pouvez spécifier différents paramètres pour chaque règle de réplication, comme suit.

```
...
```

```
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
    <Destination>
       <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
    <Destination>
       <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET2</Bucket>
    </Destination>
</Rule>
...
```

Modification du propriétaire d'un réplica

Lorsque les compartiments source et de destination ne sont pas détenus par les mêmes comptes, vous pouvez remplacer le propriétaire de la réplique par le Compte AWS propriétaire du compartiment de destination. Pour ce faire, ajoutez l'élément `AccessControlTranslation`. Cet élément prend la valeur `Destination`.

```
...
<Destination>
   <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
   <Account>destination-bucket-owner-account-id</Account>
   <AccessControlTranslation>
       <Owner>Destination</Owner>
   </AccessControlTranslation>
</Destination>
...
```

Si vous n'ajoutez pas l'`AccessControlTranslation`élément à la configuration de réplication, les répliques appartiennent au même propriétaire Compte AWS que l'objet source. Pour plus d'informations, consultez Modification du propriétaire d'un réplica.

Activer le contrôle du délai de réplication S3

Vous pouvez activer le contrôle du délai de réplication S3 dans votre configuration de réplication. S3 RTC réplique la plupart des objets en quelques secondes et 99,99 % des objets en l'espace de 15 minutes (soutenu par un contrat de niveau de service (SLA)).

> **Note**
>
> Seule une valeur de `<Minutes>15</Minutes>` est acceptée pour `EventThreshold` et `Time`.

```
...
<Destination>
  <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
```

```
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
```

Pour plus d'informations, consultez Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC). Pour des exemples d'API, consultez PutBucketReplicationle manuel Amazon Simple Storage Service API Reference.

Répliquez les objets créés avec le chiffrement côté serveur en utilisant AWS KMS

Votre compartiment source peut contenir des objets créés avec un chiffrement côté serveur à l'aide de clés AWS Key Management Service (AWS KMS) (SSE-KMS). Par défaut, Simple Storage Service (Amazon S3) ne réplique pas ces objets. Vous pouvez éventuellement demander à Simple Storage Service (Amazon S3) de répliquer ces objets. Pour ce faire, commencez par choisir explicitement cette fonctionnalité en ajoutant l'élément `SourceSelectionCriteria`. Indiquez ensuite le AWS KMS key (pour le compartiment Région AWS de destination) à utiliser pour chiffrer les répliques d'objets. Les exemples suivants montrent comment spécifier ces éléments.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Pour plus d'informations, consultez Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Exemples de configuration de réplication

Pour commencer, vous pouvez ajouter les exemples de configuration de réplication suivants à votre compartiment, selon les besoins.

> **Important**
>
> Pour ajouter une configuration de réplication à un compartiment, vous devez disposer de l'autorisation `iam:PassRole`. Cette autorisation vous permet de transmettre le rôle IAM qui accorde les autorisations de réplication à Amazon S3. Vous spécifiez le rôle IAM en fournissant l'Amazon Resource Name (ARN) utilisé dans l'élément `Role` du fichier XML de la configuration de réplication. Pour plus d'informations, consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un Service AWS dans le Guide de l'utilisateur IAM.

Example 1 : Configuration de réplication à une seule règle

La configuration de réplication de base suivante indique une règle. Cette règle spécifie un rôle IAM qu'Simple Storage Service (Amazon S3) peut endosser et un compartiment de destination unique pour les réplicas d'objets. La valeur `Status` de `Enabled` indique que la règle est en vigueur.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Pour choisir un sous-ensemble d'objets à répliquer, vous pouvez ajouter un filtre. Dans la configuration suivante, le filtre spécifie un préfixe de clé d'objet. Cette règle s'applique aux objets dotés du préfixe *Tax/* dans leur nom de clé.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
```

```
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
       <Prefix>Tax/</Prefix>
    </Filter>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Si vous spécifiez l'élément `Filter`, vous devez également inclure les éléments `Priority` et `DeleteMarkerReplication`. Dans cet exemple, `Priority` n'a aucune importance car il n'existe qu'une seule règle.

Dans la configuration suivante, le filtre spécifie un préfixe et deux balises. La règle s'applique au sous-ensemble d'objets dotés du préfixe de clé et des balises spécifiés. Elle s'applique particulièrement aux objets dotés du préfixe *Tax/* dans leur nom de clé et des deux balises d'objets spécifiées. `Priority` ne s'applique pas du fait qu'une seule règle est disponible.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
        <And>
          <Prefix>Tax/</Prefix>
          <Tag>
             <Tag>
                <Key>tagA</Key>
                <Value>valueA</Value>
             </Tag>
          </Tag>
```

```
          <Tag>
             <Tag>
                <Key>tagB</Key>
                <Value>valueB</Value>
             </Tag>
          </Tag>
       </And>

    </Filter>

    <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

  </Rule>
</ReplicationConfiguration>
```

Vous pouvez spécifier une classe de stockage pour les réplicas d'objets, comme suit.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
       <Bucket>arn:aws:s3:::example-s3-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Vous pouvez spécifier n'importe quelle classe de stockage prise en charge par Amazon S3.

Example 2 : Configuration de réplication à deux règles

Example

Dans la configuration de réplication suivante :

- Chaque règle filtre un préfixe de clé différent afin que chaque règle s'applique à un sous-ensemble distinct d'objets. Dans cet exemple, Simple Storage Service (Amazon S3) réplique les objets avec les noms de clé *Tax/doc1.pdf* et *Project/project1.txt*, mais ne réplique pas les objets avec le nom de clé *PersonalDoc/documentA*.

- La priorité des règles n'a aucune importance car les règles s'appliquent à deux ensembles d'objets distincts. L'exemple suivant décrit ce qui se passe lorsque la priorité des règles est appliquée.
- La deuxième règle spécifie la classe de stockage S3 Standard-Accès peu fréquent pour les réplicas d'objets. Simple Storage Service (Amazon S3) utilise la classe de stockage spécifiée pour ces réplicas d'objets.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
     ...
  </Rule>
 <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Project</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
     <StorageClass>STANDARD_IA</StorageClass>
    </Destination>
     ...
  </Rule>
```

```
</ReplicationConfiguration>
```

Example 3 : Configuration de réplication à deux règles et avec des préfixes qui se chevauchent

Dans cette configuration, les deux règles indiquent des filtres avec des préfixes de clé se chevauchant, *star/* et *starship/*. Les deux règles s'appliquent aux objets portant le nom de clé *starship-x*. Dans ce cas, Amazon S3 utilise la priorité des règles pour déterminer la règle à appliquer. Plus le nombre est élevé, plus la priorité est haute.

```
<ReplicationConfiguration>

  <Role>arn:aws:iam::account-id:role/role-name</Role>

  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>star</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
  </Rule>
  <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>starship</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::DOC-EXAMPLE-BUCKET1</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Example 4 : Exemples de procédures

Pour afficher des exemples, consultez la section Exemples de configuration de la réplication en direct.

Pour plus d'informations sur la structure XML de la configuration de réplication, consultez PutBucketReplicationle manuel Amazon Simple Storage Service API Reference.

Rétrocompatibilité

La dernière version du fichier XML de configuration de réplication est V2. Les configurations de réplication XML V2 sont celles qui contiennent l'élément `Filter` pour les règles et les règles qui spécifient le contrôle du délai de réplication S3.

Pour afficher votre version de configuration de réplication, vous pouvez utiliser l'opération d'API `GetBucketReplication`. Pour plus d'informations, consultez GetBucketReplicationle manuel Amazon Simple Storage Service API Reference.

Pour assurer la compatibilité descendante, Simple Storage Service (Amazon S3) continue de prendre en charge la configuration de réplication XML V1. Si vous avez utilisé la configuration de réplication XML V1, tenez compte des problèmes suivants qui ont un impact sur la compatibilité descendante :

- La version 2 du fichier XML de configuration de réplication inclut l'élément `Filter` pour les règles. L'élément `Filter` vous permet de spécifier des filtres d'objet basés sur le préfixe de clé d'objet et/ou des balises pour définir la portée des objets auxquels la règle s'applique. Le format de configuration de réplication XML V1 prend en charge le filtrage basé uniquement sur le préfixe de la clé. Dans ce cas, vous ajoutez directement le `Prefix` en tant qu'élément enfant de l'élément `Rule` comme dans l'exemple suivant.

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Role>arn:aws:iam::account-id:role/role-name</Role>
    <Rule>
      <Status>Enabled</Status>
      <Prefix>key-prefix</Prefix>
      <Destination><Bucket>arn:aws:s3:::example-s3-bucket</Bucket></Destination>

    </Rule>
  </ReplicationConfiguration>
  ```

  Pour assurer la compatibilité descendante, prend toujours en charge la configuration V1.

- Lorsque vous supprimez un objet de votre compartiment source, sans spécifier d'ID de version d'objet, Simple Storage Service (Amazon S3) ajoute un marqueur de suppression. Si vous utilisez la version 1 du fichier XML de configuration de réplication, Simple Storage Service (Amazon S3) réplique les marqueurs de suppression créés par les actions utilisateur. En d'autres termes, Amazon S3 réplique le marqueur de suppression uniquement si un utilisateur supprime un objet. Si un objet ayant expiré est supprimé par Amazon S3 (dans le cadre d'une action de cycle de vie), Amazon S3 ne réplique pas le marqueur de suppression.

  Dans les configurations de réplication V2, vous pouvez activer la réplication des marqueurs de suppression pour non-tag-based les règles. Pour plus d'informations, consultez Répliquer des marqueurs de suppression entre les compartiments.

Configuration des autorisations pour la réplication en direct

Lorsque vous configurez la réplication en direct, vous devez obtenir les autorisations nécessaires comme suit :

- Simple Storage Service (Amazon S3) a besoin d'autorisations pour répliquer des objets en votre nom. Vous accordez ces autorisations en créant un rôle IAM, puis en spécifiant ce rôle dans votre configuration de réplication.
- Lorsque les compartiments source et de destination n'appartiennent pas aux mêmes comptes, le propriétaire du compartiment de destination doit accorder au propriétaire du compartiment source les autorisations adéquates pour stocker les réplicas.

Rubriques

- Création d'un rôle IAM
- Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS
- Octroi d'autorisations pour les opérations par lots S3
- Modification du propriétaire d'un réplica
- Activer la réception d'objets répliqués à partir d'un compartiment source

Création d'un rôle IAM

Par défaut, toutes les ressources Simple Storage Service (Amazon S3) (compartiments, objets et sous-ressources liées) sont privées : seul le propriétaire des ressources peut y accéder. Simple Storage Service (Amazon S3) a besoin d'autorisations pour lire et répliquer les objets du compartiment source. Vous accordez ces autorisations en créant un rôle IAM et en spécifiant ce rôle dans votre configuration de réplication.

Cette section décrit la stratégie d'approbation et la stratégie d'autorisation minimale requise. Les exemples de procédures pas à pas fournissent des step-by-step instructions pour créer un rôle IAM. Pour plus d'informations, consultez Exemples de configuration de la réplication en direct.

- L'exemple suivant illustre une politique d'approbation selon laquelle vous identifiez Simple Storage Service (Amazon S3) en tant que principal de service capable d'endosser le rôle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- L'exemple suivant illustre une politique d'approbation selon laquelle vous identifiez Amazon S3 et les opérations par lot S3 en tant que principaux de service. Cela est utile si vous créez une tâche de réplication par lot. Pour plus d'informations, consultez Créer une tâche de réplication par lot pour une première règle de réplication ou une nouvelle destination.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service": [
              "s3.amazonaws.com",
              "batchoperations.s3.amazonaws.com"
           ]
```

```
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Pour plus d'informations sur les rôles IAM, consultez Rôles IAM dans le Guide de l'utilisateur IAM.

- L'exemple suivant illustre une stratégie d'accès selon laquelle vous accordez au rôle les autorisations lui permettant d'effectuer les tâches de réplication en votre nom. Quand Simple Storage Service (Amazon S3) endosse le rôle, il dispose des autorisations que vous avez spécifiées dans cette stratégie. Dans cette politique, *example-s3-bucket1* est le compartiment source et *example-s3-bucket2*, le compartiment de destination.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-bucket1"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-bucket1/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
```

```
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket2/*"
      }
   ]
}
```

La stratégie d'accès octroie les autorisations pour les actions suivantes :

- `s3:GetReplicationConfiguration` et `s3:ListBucket` –Les autorisations d'exécuter ces actions sur le compartiment *`example-s3-bucket1`* (compartiment source) permettent à Amazon S3 de récupérer la configuration de réplication et de répertorier le contenu du compartiment. (Le modèle d'autorisations actuel nécessite l'autorisation `s3:ListBucket` pour accéder aux marqueurs de suppression.)
- `s3:GetObjectVersionForReplication` et `s3:GetObjectVersionAcl` – Les autorisations d'effectuer ces actions accordées sur tous les objets permettent à Simple Storage Service (Amazon S3) d'obtenir une version d'objet particulière et la liste de contrôle d'accès (ACL) associée aux objets.

- `s3:ReplicateObject` et `s3:ReplicateDelete` – Les autorisations d'exécuter ces actions sur tous les objets du compartiment *`example-s3-bucket2`* (compartiment cible) permettent à Amazon S3 de répliquer des objets ou des marqueurs de suppression dans le compartiment de destination. Pour en savoir plus sur les marqueurs de suppression, consultez Impact des opérations de suppression sur la réplication.

  > **Note**
  >
  > Les autorisations relatives à l'action `s3:ReplicateObject` sur le compartiment *`example-s3-bucket2`* (le compartiment de destination) permettent également la réplication de métadonnées telles que les balises d'objet et les listes ACL. Il n'est donc pas nécessaire d'accorder une autorisation explicite pour l'action `s3:ReplicateTags`.

- `s3:GetObjectVersionTagging` – Les autorisations d'exécuter cette action sur des objets du compartiment *`example-s3-bucket1`* (compartiment source) permettent à Amazon S3 de lire les balises d'objets pour la réplication. Pour plus d'informations, consultez Catégorisation de votre stockage à l'aide de balises. Si Simple Storage Service (Amazon S3) ne dispose pas de ces autorisations, il réplique les objets mais pas leurs balises.

Pour obtenir la liste des actions Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference.

> **Important**
>
> Le Compte AWS titulaire du rôle IAM doit disposer des autorisations pour les actions qu'il accorde au rôle IAM.
> Supposons, par exemple, que le compartiment source contient des objets détenus par un autre Compte AWS. Le propriétaire des objets doit explicitement accorder à Compte AWS celui qui détient le rôle IAM les autorisations requises par le biais de l'ACL de l'objet. Dans le cas contraire, Amazon S3 ne peut pas accéder aux objets et la réplication des objets échoue. Pour plus d'informations sur les autorisations ACL, consultez Présentation de la liste de contrôle d'accès (ACL).
> Les autorisations décrites dans la présente section sont liées à la configuration de réplication minimale. Si vous choisissez d'ajouter des configurations de réplication facultatives, vous devez accorder des autorisations supplémentaires à Simple Storage Service (Amazon S3).

Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS

Lorsque les compartiments source et de destination n'appartiennent pas aux mêmes comptes, le propriétaire du compartiment de destination doit également ajouter une politique de compartiment pour accorder au propriétaire du compartiment source les autorisations adéquates pour effectuer des actions de réplication, comme suit : Dans cette politique, *example-s3-bucket2* est le compartiment de destination.

> **Note**
>
> Le format ARN du rôle peut sembler différent. Si le rôle a été créé à l'aide de la console, le format ARN est`arn:aws:iam::`*account-ID*`:role/service-role/`*role-name*. Si le rôle a été créé à l'aide du AWS CLI, le format ARN est`arn:aws:iam::`*account-ID*`:role/`*role-name*. Pour de plus amples informations, veuillez consulter IAM roles (français non garanti) dans le Guide de l'utilisateur IAM.

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3:ReplicateDelete",
            "s3:ReplicateObject"
         ],
         "Resource":"arn:aws:s3:::example-s3-bucket2/*"
      },
      {
         "Sid":"Permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action": [
            "s3:List*",
            "s3:GetBucketVersioning",
            "s3:PutBucketVersioning"
         ],
         "Resource":"arn:aws:s3::example-s3-bucket2"
      }
   ]
}
```

Pour voir un exemple, consultez Configuration d'une réplication quand les compartiments source et de destination appartiennent à des comptes distincts.

Si des objets stockés dans le compartiment source sont balisés, notez les points suivants :

- Si le propriétaire du compartiment source octroie à Amazon S3 l'autorisation d'effectuer les actions `s3:GetObjectVersionTagging` et `s3:ReplicateTags` en vue de répliquer des balises

d'objets (via le rôle IAM), Amazon S3 réplique les balises en même temps que les objets. Pour obtenir des informations sur le rôle IAM, consultez Création d'un rôle IAM.

- Si le propriétaire du compartiment de destination ne souhaite pas répliquer les balises, il peut ajouter l'instruction suivante à la stratégie du compartiment de destination en vue de lui refuser explicitement l'autorisation d'exécuter l'action `s3:ReplicateTags`. Dans cette politique, *`example-s3-bucket2`* est le compartiment de destination.

```
...
   "Statement":[
      {
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-id:role/service-role/source-account-IAM-role"
         },
         "Action":"s3:ReplicateTags",
         "Resource":"arn:aws:s3:::example-s3-bucket2/*"
      }
   ]
...
```

Octroi d'autorisations pour les opérations par lots S3

S3 Batch Replication vous permet de répliquer des objets qui existaient avant la mise en place d'une configuration de réplication, des objets qui ont déjà été répliqués et des objets dont la réplication a échoué. Vous pouvez créer une tâche de réplication par lot unique lors de la création de la première règle dans une nouvelle configuration de réplication ou de l'ajout d'une nouvelle destination à une configuration existante via la AWS Management Console. Vous pouvez également lancer la réplication par lot pour une configuration de réplication existante en créant une tâche d'opérations par lot.

Pour des exemples de politiques et de rôles IAM pour la réplication par lot, consultez Configuration des politiques IAM pour la réplication par lot.

Modification du propriétaire d'un réplica

Si différents Comptes AWS propriétaires des compartiments source et de destination sont propriétaires, vous pouvez demander à Amazon S3 de remplacer le propriétaire de la réplique par

Compte AWS celui qui détient le compartiment de destination. Pour plus d'informations sur l'option de substitution du propriétaire, consultez Modification du propriétaire d'un réplica.

Activer la réception d'objets répliqués à partir d'un compartiment source

Vous pouvez rapidement générer les politiques nécessaires pour permettre la réception d'objets répliqués à partir d'un compartiment source via la AWS Management Console.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le compartiment que vous souhaitez utiliser comme compartiment de destination.
4. Choisissez l'onglet Management (Gestion), puis faites défiler jusqu'à Replication rules (Règles de réplication).
5. Pour Actions, choisissez Receive replicated objects (Recevoir des objets répliqués).

   Suivez les instructions, entrez l' Compte AWS ID du compte du compartiment source et choisissez Generate policies. Cela générera une politique de compartiment Amazon S3 et une politique de clé KMS.
6. Pour ajouter cette politique à votre politique de compartiment existante, choisissez Apply settings (Appliquer les paramètres) ou Copy (Copier) pour copier manuellement les modifications.
7. (Facultatif) Copiez la AWS KMS politique dans la politique de clé KMS de votre choix sur la AWS Key Management Service console.

Exemples de configuration de la réplication en direct

Les exemples suivants montrent comment configurer une réplication en direct pour des cas d'utilisation courants.

> **Note**
>
> La réplication en direct fait référence à la réplication dans une même région (SRR) et une réplication entre régions (CRR). La réplication dynamique ne réplique aucun objet qui existait dans le compartiment avant que vous ne configuriez la réplication. Pour répliquer des objets qui existaient avant de configurer la réplication, utilisez la réplication à la demande. Pour

synchroniser des buckets et répliquer des objets existants à la demande, voir. Réplication d'objets existants

Ces exemples montrent comment créer une configuration de réplication à l'aide de la console Amazon S3 AWS Command Line Interface (AWS CLI) et des AWS kits de développement logiciel (AWS SDK for Java des AWS SDK for .NET exemples sont présentés).

Pour plus d'informations sur l'installation et la configuration du AWS CLI, consultez les rubriques suivantes du Guide de AWS Command Line Interface l'utilisateur.

- Installation du AWS Command Line Interface
- Configuration du AWS CLI — Vous devez configurer au moins un profil. Si vous explorez des scénarios entre comptes, configurez deux profils.

Pour plus d'informations sur AWS les SDK, consultez AWS SDK pour Java AWS et SDK pour .NET.

Tip

Pour un step-by-step didacticiel expliquant comment utiliser la réplication en direct pour répliquer des données, voir Tutoriel : Réplication de données dans et entre les deux à Régions AWS l'aide de S3 Replication.

Rubriques

- Configuration d'une réplication pour des compartiments source et destination appartenant au même compte
- Configuration d'une réplication quand les compartiments source et de destination appartiennent à des comptes distincts
- Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC)
- Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)
- Répliquer les modifications de métadonnées avec la synchronisation des modifications de réplica Amazon S3
- Répliquer des marqueurs de suppression entre les compartiments

Configuration d'une réplication pour des compartiments source et destination appartenant au même compte

La réplication est la copie automatique et asynchrone d'objets dans des compartiments identiques ou différents. Régions AWS Elle réplique les objets nouvellement créés et les mises à jour d'objets d'un compartiment source vers un ou plusieurs compartiments de destination. Pour plus d’informations, consultez Vue d'ensemble de la réplication d'objets.

Lorsque vous configurez la réplication, vous ajoutez des règles de réplication au compartiment source. Les règles de réplication définissent les objets du compartiment source à répliquer, ainsi que le ou les compartiments de destination dans lesquels les objets répliqués seront stockés. Vous pouvez créer une règle pour répliquer tous les objets ou un sous-ensemble d'objets d'un compartiment à l'aide de préfixes de nom de clé ou d'autres balises d'objet, ou les deux. Un compartiment de destination peut se trouver dans le même compartiment Compte AWS que le compartiment source ou dans un autre compte.

Si vous spécifiez un ID de version d'objet à supprimer, Amazon S3 supprime cette version de l'objet dans le compartiment source. Mais le service ne réplique pas la suppression dans le compartiment de destination. En d'autres termes, il ne supprime pas la même version de l'objet dans le compartiment de destination. Les données sont ainsi protégées contre les suppressions malencontreuses.

Lorsque vous ajoutez une règle de réplication à un compartiment, celle-ci est activée par défaut et entre en fonctionnement dès que vous l'enregistrez.

Dans cet exemple, vous configurez la réplication pour les compartiments source et de destination qui appartiennent au même Compte AWS. Des exemples d'utilisation de la console Amazon S3, du AWS Command Line Interface (AWS CLI) et du AWS SDK for Java and sont fournis AWS SDK for .NET.

Utilisation de la console S3

Pour configurer une règle de réplication lorsque le compartiment de destination se trouve dans le même compartiment Compte AWS que le compartiment source, procédez comme suit.

Si le compartiment de destination se trouve dans un compte différent du compartiment source, vous devez ajouter une stratégie de compartiment au compartiment de destination pour accorder au propriétaire du compte du compartiment source l'autorisation d'effectuer des réplications d'objets dans le compartiment de destination. Pour plus d’informations, consultez Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment que vous souhaitez utiliser.
4. Sélectionnez Gestion, faites défiler jusqu'à Règles de réplication, puis sélectionnez Créer une règle de réplication.

5. Dans la section Configuration de la règle de réplication, sous Nom de la règle de réplication, saisissez un nom pour votre règle afin de l'identifier facilement plus tard. Ce nom est obligatoire et doit être unique dans le compartiment.
6. Sous Statut, Activé est sélectionné par défaut. Un règle activée entre en fonctionnement dès que l'avez enregistrée. Si vous souhaitez activer la règle ultérieurement, sélectionnez Désactivé.
7. Si le compartiment possède des règles de réplication existantes, il vous est demandé de définir une priorité pour la règle. Vous devez définir une priorité pour la règle pour éviter les conflits provoqués par les objets inclus dans l'étendue de plusieurs règles. En cas de chevauchement de règles, Amazon S3 utilise la priorité des règles pour déterminer la règle à appliquer. Plus le nombre est élevé, plus la priorité est haute. Pour plus d'informations sur la priorité des règles, consultez Configuration de réplication.
8. Sous Compartiment source, vous disposez des options suivantes pour définir la source de réplication :

   - Pour répliquer l'ensemble du compartiment, choisissez Apply to all objects in the bucket (Appliquer à tous les objets du compartiment).
   - Pour répliquer tous les objets ayant le même préfixe, choisissez Limit the scope of this rule using one or more filters (Limiter la portée de cette règle en utilisant un ou plusieurs filtres). Cela limite la réplication à tous les objets dont les noms commencent par le préfixe que vous spécifiez (par exemple, `pictures`). Saisissez un préfixe dans la zone Préfixe.

     > **Note**
     >
     > Si vous entrez un préfixe correspondant à un nom de dossier, vous devez insérer le caractère / (barre oblique) à la fin (par exemple, `pictures/`).

   - Pour répliquer tous les objets avec une ou plusieurs balises d'objet, sélectionnez Ajouter une balise et saisissez la paire clé-valeur dans les zones. Répétez la procédure pour ajouter une

autre balise. Vous pouvez combiner un préfixe et des balises. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.

Le nouveau schéma XML de configuration de la réplication prend en charge le balisage de préfixe et de balise, et la priorité des règles. Pour plus d'informations sur le nouveau schéma, consultez Rétrocompatibilité. Pour plus d'informations sur le XML utilisé avec l'API Amazon S3 qui fonctionne derrière l'interface utilisateur, reportez-vous à Configuration de réplication. Le nouveau schéma est décrit comme configuration de réplication XML V2.

9. Sous Destination, sélectionnez le compartiment où vous souhaitez qu'Amazon S3 réplique les objets.

> **Note**
>
> Le nombre de compartiments de destination est limité au nombre de compartiments contenus Régions AWS dans une partition donnée. Une partition est un regroupement de régions. AWS possède actuellement trois partitions : `aws` (Régions standard), `aws-cn` (Régions de Chine) et `aws-us-gov` (AWS GovCloud (US) Régions). Vous pouvez utiliser Service Quotas pour demander une augmentation de votre limite de compartiments de destination.

- Pour répliquer vers un compartiment ou plusieurs compartiments de votre compte, sélectionnez Choisir un compartiment dans ce compte et saisissez ou recherchez le nom du compartiment de destination.
- Pour effectuer une réplication vers un ou plusieurs compartiments d'un autre compte Compte AWS, choisissez Spécifier un compartiment dans un autre compte, puis entrez l'ID du compte du compartiment de destination et le nom du compartiment.

  Si la destination se trouve dans un compte différent du compartiment source, vous devez ajouter une stratégie de compartiment aux compartiments de destination pour accorder au propriétaire du compte du compartiment source l'autorisation d'effectuer des réplications d'objets. Pour plus d'informations, consultez Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS.

  (Facultatif) Si vous souhaitez aider à normaliser la propriété des nouveaux objets dans le compartiment de destination, choisissez Remplacer la propriété de l'objet par le propriétaire

du compartiment de destination. Pour plus d'informations sur cette option, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Note**
>
> Si la gestion des versions n'est pas activée sur le compartiment de destination, un message d'avertissement avec le bouton Activer la gestion des versions s'affiche. Cliquez sur ce bouton pour activer la gestion des versions sur le compartiment.

10. Configurez un rôle AWS Identity and Access Management (IAM) qu'Amazon S3 peut assumer pour répliquer des objets en votre nom.

    Pour configurer un rôle IAM, dans la section Rôle IAM, sélectionnez l'une des options suivantes dans la liste déroulante des rôles IAM :

    - Nous vous recommandons vivement de choisir Create new role (Créer un nouveau rôle) pour demander à Amazon S3 de créer nouveau rôle IAM pour vous. Lorsque vous enregistrez la règle, une nouvelle stratégie est générée pour le rôle IAM correspondant aux compartiments source et cible que vous choisissez.
    - Vous pouvez également choisir d'utiliser un rôle IAM existant. Dans ce cas, vous devez choisir un rôle qui octroie à Amazon S3 les autorisations nécessaires pour la réplication. La réplication échoue si ce rôle n'accorde pas à Amazon S3 les autorisations suffisantes pour suivre votre règle de réplication.

    > **Important**
    >
    > Lorsque vous ajoutez une règle de réplication à un compartiment, vous devez disposer de l'autorisation `iam:PassRole` pour pouvoir transmettre le rôle IAM qui accorde les autorisations de réplication Amazon S3. Pour plus d'informations, consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un Service AWS dans le Guide de l'utilisateur IAM.

11. Pour répliquer les objets du compartiment source chiffrés à l'aide du chiffrement côté serveur à l'aide de clés AWS Key Management Service (AWS KMS) (SSE-KMS), sous Chiffrement, sélectionnez Répliquer les objets chiffrés avec. AWS KMS Sous Clés AWS KMS pour le chiffrement des objets de destination se trouvent les clés source que la réplication est autorisée

à utiliser. Toutes les clés source KMS sont incluses par défaut. Vous pouvez choisir un alias ou un ID de clé afin de restreindre la sélection des clés KMS.

Les objets chiffrés par AWS KMS keys ceux que vous ne sélectionnez pas ne sont pas répliqués. Une clé KMS ou un groupe de clés KMS est sélectionné pour vous, mais vous pouvez choisir les clés KMS que vous souhaitez utiliser si vous le souhaitez. Pour plus d'informations sur l'utilisation AWS KMS avec la réplication, consultezRéplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

> **Important**
>
> Lorsque vous répliquez des objets chiffrés avec AWS KMS, le taux de AWS KMS demandes double dans la région source et augmente d'autant dans la région de destination. Ces taux d'appels accrus AWS KMS sont dus à la manière dont les données sont rechiffrées à l'aide de la clé KMS que vous définissez pour la région de destination de la réplication. AWS KMS dispose d'un quota de taux de demandes par compte d'appel et par région. Pour obtenir des informations sur les quotas par défaut, consultez Quotas AWS KMS – nombre de demandes par seconde : variable dans le Guide du développeur AWS Key Management Service .
>
> Si votre taux actuel de demandes d'PUTobjets Amazon S3 pendant la réplication est supérieur à la moitié de la limite de AWS KMS débit par défaut de votre compte, nous vous recommandons de demander une augmentation de votre quota de taux de AWS KMS demandes. Pour demander une augmentation, contactez-nous afin de créer un cas dans le Centre AWS Support . Supposons, par exemple, que votre taux de demandes d'PUTobjets actuel soit de 1 000 requêtes par seconde et que vous l'utilisiez AWS KMS pour chiffrer vos objets. Dans ce cas, nous vous recommandons de demander AWS Support à augmenter votre limite de AWS KMS débit à 2 500 requêtes par seconde, à la fois dans vos régions source et de destination (si elles sont différentes), afin de garantir qu'il n'y ait pas de limitation. AWS KMS
>
> Pour connaître le taux de demandes d'PUTobjets dans le compartiment source, consultez `PutRequests` les métriques des CloudWatch demandes Amazon pour Amazon S3. Pour plus d'informations sur l'affichage CloudWatch des métriques, consultezUtiliser la console S3..

Si vous avez choisi de répliquer des objets chiffrés avec AWS KMS, procédez comme suit :

- Sous AWS KMS key pour le chiffrement des objets de destination, spécifiez votre clé KMS de l'une des manières suivantes :

  - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos clés AWS KMS keys, puis sélectionnez votre Clé KMS dans la liste des clés disponibles.

    La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .

  - Pour saisir l'Amazon Resource Name (ARN) de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche. Cela chiffre les réplicas dans le compartiment de destination. Vous trouverez l'ARN de votre clé KMS dans la console IAM, sous Clés de chiffrement.

  - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

    Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

    > **Important**
    >
    > Vous ne pouvez utiliser que les clés KMS activées au même endroit Région AWS que le bucket. Lorsque vous choisissez Choisir parmi vos clés KMS, la console S3 ne répertorie que 100 clés KMS par région. Si vous avez plus de 100 clés KMS dans la même Région, vous pourrez uniquement afficher les 100 premières clés KMS dans la console S3. Pour utiliser une clé KMS qui n'est pas répertoriée dans la console, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de la clé KMS.
    > Lorsque vous utilisez un AWS KMS key pour le chiffrement côté serveur dans Amazon S3, vous devez choisir une clé KMS de chiffrement symétrique. Amazon S3 prend uniquement en charge les clés KMS symétriques de chiffrement et ne prend pas en charge les clés KMS asymétriques. Pour plus d'informations, consultez la section Identifying symmetric and asymmetric KMS keys (Identification des

clés KMS symétriques et asymétriques) dans le Guide du développeur AWS Key Management Service .

Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur. Pour plus d'informations sur l'utilisation AWS KMS avec Amazon S3, consultezUtilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

12. Sous Classe de stockage de destination, si vous voulez répliquer vos données dans une classe de stockage spécifique dans le compartiment de destination, sélectionnez Modifier la classe de stockage pour les objets répliqués. Choisissez ensuite la classe de stockage que vous voulez utiliser pour les objets répliqués dans la destination. Si vous ne sélectionnez pas cette option, la classe de stockage utilisée pour les objets répliqués est identique à celle des objets d'origine.
13. Vous disposez des options supplémentaires suivantes lors de la définition des Options de réplication supplémentaires :

    - Si vous souhaitez activer le Contrôle du temps de réplication S3 (S3 RTC) dans votre configuration de réplication, sélectionnez Contrôle du délai de réplication (RTC). Pour plus d'informations sur cette option, consultez Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC).
    - Si vous souhaitez activer les métriques de réplication S3 dans votre configuration de réplication, sélectionnez Replication metrics and events (Métriques et événements de réplication). Pour plus d'informations, consultez Surveillance de la progression avec des métriques de réplication et des notifications d'événements S3.
    - Si vous souhaitez activer la réplication de marqueurs de suppression dans votre configuration de réplication, sélectionnez Réplication des marqueurs de suppression. Pour plus d'informations, consultez Répliquer des marqueurs de suppression entre les compartiments.
    - Si vous souhaitez activer la synchronisation des modifications de réplica Amazon S3 dans votre configuration de réplication, sélectionnez Synchronisation des modifications de réplica. Pour plus d'informations, consultez Répliquer les modifications de métadonnées avec la synchronisation des modifications de réplica Amazon S3.

> **Note**
>
> Des frais supplémentaires s'appliquent lorsque vous utilisez des métriques de réplication S3 RTC ou S3.

14. Pour terminer, choisissez Enregistrer.
15. Une fois votre règle enregistrée, vous pouvez la modifier, l'activer, la désactiver ou la supprimer en la sélectionnant et en choisissant Edit rule (Modifier la règle).

En utilisant le AWS CLI

AWS CLI Pour configurer la réplication lorsque les compartiments source et de destination appartiennent à la même entité Compte AWS, procédez comme suit :

- Créer des compartiments source et de destination
- Activer la gestion des versions sur les compartiments
- Créer un rôle IAM qui octroie à Simple Storage Service (Amazon S3) l'autorisation de répliquer des objets
- Ajouter la configuration de réplication au compartiment source

Testez votre configuration pour la vérifier.

Pour configurer la réplication lorsque les compartiments source et de destination appartiennent à la même entité Compte AWS

1. Définissez un profil d'informations d'identification pour l' AWS CLI. Dans cet exemple, nous utilisons le nom de profil `acctA`. Pour plus d’informations sur la définition des profils d’informations d’identification, consultez Profils nommés dans le Guide de l’utilisateur AWS Command Line Interface .

   > **Important**
   >
   > Le profil que vous utilisez pour cet exercice doit disposer des autorisations nécessaires. Par exemple, dans la configuration de réplication, vous spécifiez le rôle IAM qu'Amazon S3 peut endosser. Vous ne pouvez effectuer cette tâche que si le profil que vous utilisez dispose de l'autorisation `iam:PassRole`. Pour plus d’informations, consultez Octroi

d'autorisations à un utilisateur pour transférer un rôle à un service AWS dans le Guide de l'utilisateur IAM. Si vous utilisez les informations d'identification d'un administrateur pour créer un profil nommé, vous pouvez exécuter toutes les tâches.

2. Créez un compartiment *source* et activez la gestion des versions sur ce dernier. Le code suivant crée un compartiment *source* dans la Région USA Est (Virginie du Nord) (us-east-1).

```
aws s3api create-bucket \
--bucket source \
--region us-east-1 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket source \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Créez un compartiment *destination* et activez la gestion des versions sur ce dernier. Le code suivant crée un compartiment *destination* dans la Région USA Ouest (Oregon) (us-west-2).

> **Note**
>
> Pour configurer la configuration de réplication lorsque les compartiments source et de destination se trouvent dans le même compartiment Compte AWS, vous devez utiliser le même profil. Cet exemple utilise `acctA`. Pour tester la configuration de réplication lorsque les buckets appartiennent à des propriétaires différents Comptes AWS, vous devez spécifier des profils différents pour chacun d'entre eux. Cet exemple utilise le profil `acctB` pour le compartiment de destination.

```
aws s3api create-bucket \
--bucket destination \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
```

```
--bucket destination \
--versioning-configuration Status=Enabled \
--profile acctA
```

4. Créez un rôle IAM. Vous précisez ce rôle dans la configuration de réplication que vous ajouterez ultérieurement au compartiment *source*. Amazon S3 endosse ce rôle pour répliquer des objets en votre nom. Vous créez un rôle IAM en deux étapes.

   - Créez un rôle
   - Attachez une stratégie d'autorisation au rôle.

   a. Créez le rôle IAM.

      i. Copiez la stratégie d'approbation suivante et enregistrez-la dans un fichier nommé `s3-role-trust-policy.json` dans le répertoire actif sur votre ordinateur local. Cette stratégie accorde au principal de service Amazon S3 les autorisations principales pour endosser ce rôle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Exécutez la commande suivante pour créer un rôle.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

   b. Attachez une stratégie d'autorisation au rôle.

i. Copiez la politique d'autorisations suivante et enregistrez-la dans un fichier nommé `s3-role-permissions-policy.json` dans le répertoire actuel de votre ordinateur local. Cette stratégie accorde des autorisations pour diverses actions sur les compartiments et les objets Amazon S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::destination-bucket/*"
      }
   ]
}
```

ii. Exécutez la commande suivante pour créer une stratégie et l'attacher au rôle.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Ajoutez une configuration de réplication au compartiment *source*.

   a. Bien que l'API Amazon S3 nécessite une configuration de réplication au AWS CLI format XML, vous devez spécifier la configuration de réplication au format JSON. Enregistrez la configuration JSON dans un fichier (`replication.json`) dans le répertoire local de votre ordinateur.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::destination-bucket"
      }
    }
  ]
}
```

   b. Mettez à jour le fichier JSON en fournissant des valeurs pour *destination-bucket* et *IAM-role-ARN*. Enregistrez les modifications.

   c. Pour ajouter la configuration de réplication à votre compartiment source, exécutez la commande suivante. Veillez à saisir le nom du compartiment *source*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
--profile acctA
```

Pour récupérer la configuration de réplication, utilisez la commande `get-bucket-replication`.

```
$ aws s3api get-bucket-replication \
--bucket source \
--profile acctA
```

6. Testez la configuration dans la console Amazon S3 :

   a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

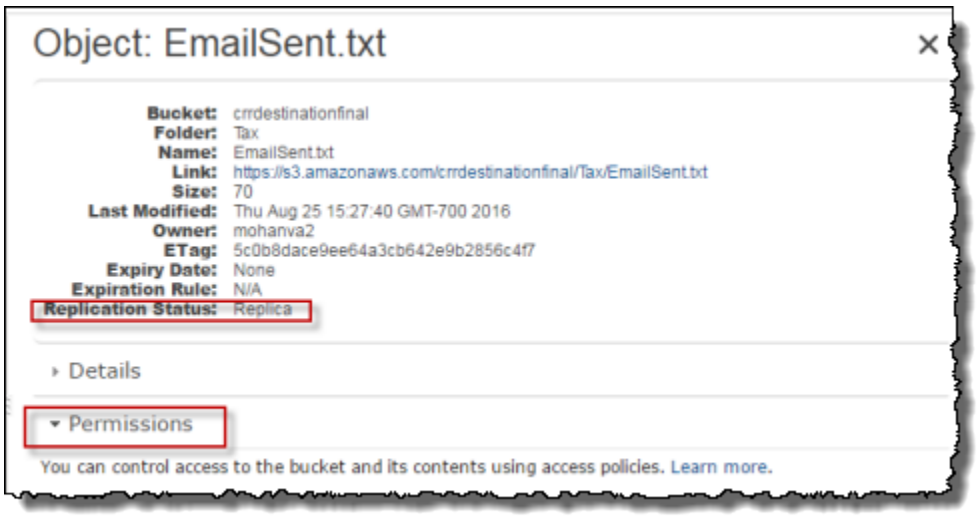
   b. Dans le compartiment *source*, créez un dossier nommé `Tax`.

   c. Ajoutez les exemples d'objets au dossier `Tax` du compartiment *source*.

   > **Note**
   >
   > Le temps nécessaire à Simple Storage Service (Amazon S3) pour répliquer un objet dépend de la taille de ce dernier. Pour obtenir des informations sur la consultation du statut de la réplication, consultez Obtention d'informations sur le statut de la réplication.

   Dans le compartiment *destination*, vérifiez les éléments suivants :

   - Simple Storage Service (Amazon S3) a répliqué les objets.
   - Dans les propriétés de l'objet, le Statut de réplication est défini avec la valeur `Replica` (identifiant celui-ci comme objet de réplication).
   - Dans les propriétés de l'objet, la section des autorisations n'affiche aucune autorisation. Cela signifie que le réplica continue d'être la propriété du propriétaire du compartiment *source* et que le propriétaire du compartiment *destination* n'a aucune autorisation sur le réplica de l'objet. Vous pouvez ajouter une configuration facultative pour indiquer à Simple Storage Service (Amazon S3) de modifier le propriétaire du réplica. Pour obtenir un exemple, consultez Comment changer le propriétaire de la réplique.



Utilisation des AWS SDK

Utilisez les exemples de code suivants pour ajouter une configuration de réplication à un compartiment avec le AWS SDK for Java et AWS SDK for .NET, respectivement.

Java

L'exemple suivant ajoute une configuration de réplication à un compartiment, puis la récupère et la vérifie. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagement;
import
 com.amazonaws.services.identitymanagement.AmazonIdentityManagementClientBuilder;
import com.amazonaws.services.identitymanagement.model.CreateRoleRequest;
import com.amazonaws.services.identitymanagement.model.PutRolePolicyRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.CreateBucketRequest;
```

```
import com.amazonaws.services.s3.model.DeleteMarkerReplication;
import com.amazonaws.services.s3.model.DeleteMarkerReplicationStatus;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.replication.ReplicationFilter;
import com.amazonaws.services.s3.model.replication.ReplicationFilterPredicate;
import com.amazonaws.services.s3.model.replication.ReplicationPrefixPredicate;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CrossRegionReplication {

        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String accountId = "*** Account ID ***";
                String roleName = "*** Role name ***";
                String sourceBucketName = "*** Source bucket name ***";
                String destBucketName = "*** Destination bucket name ***";
                String prefix = "Tax/";

                String roleARN = String.format("arn:aws:iam::%s:%s", accountId,
 roleName);
                String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

                AmazonS3 s3Client = AmazonS3Client.builder()
                                .withCredentials(new ProfileCredentialsProvider())
                                .withRegion(clientRegion)
                                .build();

                createBucket(s3Client, clientRegion, sourceBucketName);
                createBucket(s3Client, clientRegion, destBucketName);
                assignRole(roleName, clientRegion, sourceBucketName,
 destBucketName);

                try {

                        // Create the replication rule.
```

```
                        List<ReplicationFilterPredicate> andOperands = new
ArrayList<ReplicationFilterPredicate>();
                        andOperands.add(new ReplicationPrefixPredicate(prefix));

                        Map<String, ReplicationRule> replicationRules = new
HashMap<String, ReplicationRule>();
                        replicationRules.put("ReplicationRule1",
                                        new ReplicationRule()
                                                        .withPriority(0)

.withStatus(ReplicationRuleStatus.Enabled)

.withDeleteMarkerReplication(
                                                                        new
DeleteMarkerReplication().withStatus(

   DeleteMarkerReplicationStatus.DISABLED))
                                                        .withFilter(new
ReplicationFilter().withPredicate(
                                                                        new
ReplicationPrefixPredicate(prefix)))
                                                        .withDestinationConfig(new
ReplicationDestinationConfig()

.withBucketARN(destinationBucketARN)

.withStorageClass(StorageClass.Standard)));

                        // Save the replication rule to the source bucket.
                        s3Client.setBucketReplicationConfiguration(sourceBucketName,
                                        new BucketReplicationConfiguration()
                                                        .withRoleARN(roleARN)

.withRules(replicationRules));

                        // Retrieve the replication configuration and verify that
the configuration
                        // matches the rule we just set.
                        BucketReplicationConfiguration replicationConfig = s3Client

.getBucketReplicationConfiguration(sourceBucketName);
                        ReplicationRule rule =
replicationConfig.getRule("ReplicationRule1");
                        System.out.println("Retrieved destination bucket ARN: "
```

```
                                        +
rule.getDestinationConfig().getBucketARN());
                        System.out.println("Retrieved priority: " +
rule.getPriority());
                        System.out.println("Retrieved source-bucket replication rule
status: " + rule.getStatus());
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
        }

        private static void createBucket(AmazonS3 s3Client, Regions region, String
bucketName) {
                CreateBucketRequest request = new CreateBucketRequest(bucketName,
region.getName());
                s3Client.createBucket(request);
                BucketVersioningConfiguration configuration = new
BucketVersioningConfiguration()
                                .withStatus(BucketVersioningConfiguration.ENABLED);

                SetBucketVersioningConfigurationRequest enableVersioningRequest =
new SetBucketVersioningConfigurationRequest(
                                bucketName, configuration);
                s3Client.setBucketVersioningConfiguration(enableVersioningRequest);

        }

        private static void assignRole(String roleName, Regions region, String
sourceBucket, String destinationBucket) {
                AmazonIdentityManagement iamClient =
AmazonIdentityManagementClientBuilder.standard()
                                .withRegion(region)
                                .withCredentials(new ProfileCredentialsProvider())
                                .build();
                StringBuilder trustPolicy = new StringBuilder();
                trustPolicy.append("{\\r\\n   ");
```

```
                trustPolicy.append("\\\"Version\\\":\\\"2012-10-17\\\",\\r\\n   ");
                trustPolicy.append("\\\"Statement\\\":[\\r\\n      {\\r\\n
 ");
                trustPolicy.append("\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Principal\\\":{\\r\\n            ");
                trustPolicy.append("\\\"Service\\\":\\\"s3.amazonaws.com\\\"\\r\\n
      },\\r\\n         ");
                trustPolicy.append("\\\"Action\\\":\\\"sts:AssumeRole\\\"\\r\\n
  }\\r\\n   ]\\r\\n}");

                CreateRoleRequest createRoleRequest = new CreateRoleRequest()
                                .withRoleName(roleName)

 .withAssumeRolePolicyDocument(trustPolicy.toString());

                iamClient.createRole(createRoleRequest);

                StringBuilder permissionPolicy = new StringBuilder();
                permissionPolicy.append(
                                "{\\r\\n   \\\"Version\\\":\\\"2012-10-17\\\",\\r\\n
   \\\"Statement\\\":[\\r\\n      {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Action\\\":[\\r\\n             ");
                permissionPolicy.append("\\\"s3:GetObjectVersionForReplication\\\",\
\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:GetObjectVersionAcl\\\"\\r\\n         ],\\r\
\n         \\\"Resource\\\":[\\r\\n            ");
                permissionPolicy.append("\\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("/*\\\"\\r\\n         ]\\r\\n      },\\r\\n
     {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Action\\\":[\\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:ListBucket\\\",\\r\\n            \\
\"s3:GetReplicationConfiguration\\\"\\r\\n         ");
                permissionPolicy.append("],\\r\\n         \\\"Resource\\\":[\\r\\n
          \\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("\\r\\n         ");
                permissionPolicy
```

```
                                .append("]\\r\\n      },\\r\\n      {\\r\\n
 \\\"Effect\\\":\\\"Allow\\\",\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Action\\\":[\\r\\n           \\
\"s3:ReplicateObject\\\",\\r\\n           ");
                permissionPolicy
                                .append("\\\"s3:ReplicateDelete\\\",\\r\\n
  \\\"s3:ReplicateTags\\\",\\r\\n            ");
                permissionPolicy.append("\\\"s3:GetObjectVersionTagging\\\"\\r\\n\\r
\\n         ],\\r\\n         ");
                permissionPolicy.append("\\\"Resource\\\":\\\"arn:aws:s3:::");
                permissionPolicy.append(destinationBucket);
                permissionPolicy.append("/*\\\"\\r\\n      }\\r\\n   ]\\r\\n}");

                PutRolePolicyRequest putRolePolicyRequest = new
 PutRolePolicyRequest()
                                .withRoleName(roleName)
                                .withPolicyDocument(permissionPolicy.toString())
                                .withPolicyName("crrRolePolicy");

                iamClient.putRolePolicy(putRolePolicyRequest);

        }
}
```

C#

L'exemple de AWS SDK for .NET code suivant ajoute une configuration de réplication à un compartiment, puis la récupère. Pour utiliser ce code, fournissez les noms de vos compartiments et l'Amazon Resource Name (ARN) de votre rôle IAM. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
```

```
   {
       private const string sourceBucket = "*** source bucket ***";
       // Bucket ARN example - arn:aws:s3:::destinationbucket
       private const string destinationBucketArn = "*** destination bucket ARN
***";
       private const string roleArn = "*** IAM Role ARN ***";
       // Specify your bucket region (an example region is shown).
       private static readonly RegionEndpoint sourceBucketRegion =
RegionEndpoint.USWest2;
       private static IAmazonS3 s3Client;
       public static void Main()
       {
           s3Client = new AmazonS3Client(sourceBucketRegion);
           EnableReplicationAsync().Wait();
       }
       static async Task EnableReplicationAsync()
       {
           try
           {
               ReplicationConfiguration replConfig = new ReplicationConfiguration
               {
                   Role = roleArn,
                   Rules =
                       {
                           new ReplicationRule
                           {
                               Prefix = "Tax",
                               Status = ReplicationRuleStatus.Enabled,
                               Destination = new ReplicationDestination
                               {
                                   BucketArn = destinationBucketArn
                               }
                           }
                       }
               };

               PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
               {
                   BucketName = sourceBucket,
                   Configuration = replConfig
               };
```

```
                PutBucketReplicationResponse putResponse = await
 s3Client.PutBucketReplicationAsync(putRequest);

                // Verify configuration by retrieving it.
                await RetrieveReplicationConfigurationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
        private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
 client)
        {
            // Retrieve the configuration.
            GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
            {
                BucketName = sourceBucket
            };
            GetBucketReplicationResponse getResponse = await
 client.GetBucketReplicationAsync(getRequest);
            // Print.
            Console.WriteLine("Printing replication configuration information...");
            Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
            foreach (var rule in getResponse.Configuration.Rules)
            {
                Console.WriteLine("ID: {0}", rule.Id);
                Console.WriteLine("Prefix: {0}", rule.Prefix);
                Console.WriteLine("Status: {0}", rule.Status);
            }
        }
    }
}
```

Configuration d'une réplication quand les compartiments source et de destination appartiennent à des comptes distincts

La configuration de la réplication lorsque les compartiments *source* et de *destination* appartiennent à des propriétaires différents Comptes AWS est similaire à la configuration de la réplication lorsque les deux compartiments appartiennent au même compte. La seule différence est que le propriétaire du compartiment de *destination* doit accorder au propriétaire du compartiment *source* l'autorisation de répliquer des objets en ajoutant une stratégie de compartiment.

Pour plus d'informations sur la configuration de la réplication utilisant un chiffrement côté serveur avec AWS Key Management Service dans des scénarios entre comptes, consultez Octroi d'autorisations supplémentaires pour les scénarios à plusieurs comptes.

Pour configurer la réplication lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS

1. Dans cet exemple, vous créez des compartiments *source* et de *destination* dans deux compartiments différents Comptes AWS. Vous devez définir deux profils d'identification pour le AWS CLI (dans cet exemple, nous utilisons `acctA` et `acctB` pour les noms de profil). Pour plus d'informations sur la définition de profils d'informations d'identification, consultez Profils nommés dans le Guide de l'utilisateur AWS Command Line Interface .
2. Suivez les step-by-step instructions en Configuration pour des compartiments dans le même compte apportant les modifications suivantes :
   - Pour toutes les AWS CLI commandes liées aux activités du compartiment *source* (pour créer le compartiment *source*, activer le versionnement et créer le rôle IAM), utilisez le `acctA` profil. Utilisez le profil `acctB` pour créer le compartiment de *destination*.
   - Assurez-vous que la stratégie d'autorisations spécifie les compartiments *source* et de *destination* que vous avez créés pour cet exemple.
3. Dans la console, ajoutez la stratégie de compartiment suivante au compartiment de *destination* pour autoriser le propriétaire du compartiment *source* à répliquer des objets. Veillez à modifier la politique en fournissant l' Compte AWS ID du propriétaire du compartiment *source* et le nom du compartiment de *destination*.

   > **Note**
   >
   > Pour utiliser l'exemple suivant, remplacez *user input placeholders* par vos propres informations. Remplacez *DOC-EXAMPLE-BUCKET* par le nom de votre

compartiment de destination. Remplacez *source-bucket-acct-ID:role/service-role/source-acct-iam-role par le rôle* que vous utilisez pour cette configuration de réplication.
Si vous avez créé la fonction du service IAM manuellement, définissez le chemin du rôle sur `role/service-role/`, comme indiqué dans l'exemple de politique ci-dessous. Pour plus d'informations, consultez ARN IAM dans le Guide de l'utilisateur IAM.

```
{
   "Version":"2012-10-17",
   "Id":"",
   "Statement":[
      {
         "Sid":"Set-permissions-for-objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-IAM-role"
         },
         "Action":["s3:ReplicateObject", "s3:ReplicateDelete"],
         "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      },
      {
         "Sid":"Set permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-acct-ID:role/service-role/source-acct-IAM-role"
         },
         "Action":["s3:GetBucketVersioning", "s3:PutBucketVersioning"],
         "Resource":"arn:aws:s3:::DOC-EXAMPLE-BUCKET"
      }
   ]
}
```

Choisissez le compartiment et ajoutez la stratégie de compartiment. Pour obtenir des instructions, veuillez consulter Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3.

Dans une réplication, par défaut, le réplica appartient au propriétaire de l'objet source. Lorsque les compartiments source et de destination appartiennent à des propriétaires différents Comptes AWS,

vous pouvez ajouter des paramètres de configuration facultatifs pour remplacer la propriété des répliques par le Compte AWS propriétaire des compartiments de destination. Cela comprend l'octroi de l'autorisation `ObjectOwnerOverrideToBucketOwner`. Pour plus d'informations, consultez Modification du propriétaire d'un réplica.

Modification du propriétaire d'un réplica

Dans une réplication, par défaut, le réplica appartient également au propriétaire de l'objet source. Lorsque les compartiments source et de destination appartiennent à des propriétaires différents Comptes AWS et que vous souhaitez remplacer la propriété des répliques par Compte AWS celle qui détient les compartiments de destination, vous pouvez ajouter des paramètres de configuration facultatifs pour remplacer la propriété des répliques par Compte AWS celle du propriétaire des compartiments de destination. Vus pouvez choisir de le faire, par exemple, pour limiter l'accès aux réplicas d'objet. C'est ce qu'on appelle l'option de substitution du propriétaire de la configuration de réplication. Pour plus d'informations sur l'option de substitution du propriétaire, consultez Ajout de l'option de substitution du propriétaire à la configuration de réplication. Pour obtenir des informations sur la définition d'une configuration de réplication, consultez Vue d'ensemble de la réplication d'objets.

Pour configurer la substitution du propriétaire, effectuez les opérations suivantes :

- Ajoutez l'option de substitution du propriétaire à la configuration de réplication pour indiquer à Amazon S3 de modifier le propriétaire des réplicas.
- Accordez à Amazon S3 les autorisations de modifier le propriétaire des réplicas.
- Ajoutez l'autorisation dans la stratégie des compartiments de destination pour autoriser la modification de propriété du réplica. Cela permet au propriétaire des compartiments de destination d'accepter la propriété des réplicas d'objet.

Pour plus d'informations, consultez Ajout de l'option de substitution du propriétaire à la configuration de réplication. Pour un exemple pratique avec des step-by-step instructions, voirComment changer le propriétaire de la réplique.

Paramètre de propriétaire du compartiment imposé pour la propriété de l'objet

Lorsque vous utilisez la réplication Amazon S3 et que les compartiments source et de destination appartiennent à des propriétaires différents Comptes AWS, le propriétaire du compartiment de destination peut désactiver les ACL (le propriétaire du compartiment étant défini comme propriétaire de l'objet) afin de remplacer la propriété de la réplique par le Compte AWS propriétaire du

compartiment de destination. Ce paramètre imite le comportement de remplacement du propriétaire existant sans avoir besoin d'une autorisation `s3:ObjectOwnerOverrideToBucketOwner`. Cela signifie que tous les objets répliqués dans le compartiment de destination avec le paramètre bucket owner enforced (propriétaire du compartiment imposé) appartiennent au propriétaire du compartiment de destination. Pour en savoir plus sur la propriété des objets, veuillez consulter Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Ajout de l'option de substitution du propriétaire à la configuration de réplication

> **Warning**
>
> Ajoutez l'option de remplacement du propriétaire uniquement lorsque les compartiments source et de destination appartiennent à des propriétaires différents. Comptes AWS Amazon S3 ne vérifie pas si les compartiments appartiennent au même compte ou à des comptes différents. Si vous ajoutez le remplacement du propriétaire alors que les deux compartiments appartiennent au même propriétaire Compte AWS, Amazon S3 applique le remplacement du propriétaire. Il accorde les autorisations complètes au propriétaire du compartiment de destination et ne réplique pas les mises à jour ultérieures de la liste de contrôle d'accès (ACL) de l'objet source. Le propriétaire du réplica peut procéder directement à des modifications de la liste de contrôle d'accès (ACL) associée à un réplica à l'aide d'une requête `PUT ACL`, mais pas par l'intermédiaire de la réplication.

Pour spécifier l'option de substitution du propriétaire, ajoutez les informations suivantes à chaque élément `Destination` :

- L'élément `AccessControlTranslation`, qui indique à Amazon S3 de modifier le propriétaire des réplicas
- L'`Account`élément, qui indique le propriétaire Compte AWS du compartiment de destination

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    ...
    <Destination>
      ...
      <AccessControlTranslation>
           <Owner>Destination</Owner>
       </AccessControlTranslation>
      <Account>destination-bucket-owner-account-id</Account>
```

```
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

L'exemple de configuration de réplication suivant indique à Amazon S3 de répliquer les objets ayant le préfixe de clé `Tax` dans le compartiment de destination et de modifier le propriétaire des réplicas.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::destination-bucket</Bucket>
         <Account>destination-bucket-owner-account-id</Account>
         <AccessControlTranslation>
            <Owner>Destination</Owner>
         </AccessControlTranslation>
      </Destination>
   </Rule>
</ReplicationConfiguration>
```

Octroi à Amazon S3 de l'autorisation de modifier le propriétaire des réplicas

Accordez à Amazon S3 les autorisations de modifier le propriétaire des réplicas en ajoutant une autorisation pour l'action `s3:ObjectOwnerOverrideToBucketOwner` dans la stratégie d'autorisations associée au rôle IAM. Il s'agit du rôle IAM spécifié dans la configuration de réplication qui autorise Amazon S3 à endosser le rôle et à répliquer les objets en votre nom.

```
...
{
    "Effect":"Allow",
         "Action":[
       "s3:ObjectOwnerOverrideToBucketOwner"
```

```
    ],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Ajout d'autorisations dans la stratégie du compartiment de destination pour autoriser la modification de propriété du réplica

Le propriétaire du compartiment de destination doit accorder au propriétaire du compartiment source l'autorisation de modifier la propriété du réplica. Le propriétaire du compartiment de destination accorde au propriétaire du compartiment source l'autorisation pour l'action `s3:ObjectOwnerOverrideToBucketOwner`. Cela permet au propriétaire du compartiment de destination d'accepter la propriété des réplicas d'objet. L'exemple d'instruction de stratégie de compartiment suivant montre comment procéder.

```
...
{
    "Sid":"1",
    "Effect":"Allow",
    "Principal":{"AWS":"source-bucket-account-id"},
    "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource":"arn:aws:s3:::destination-bucket/*"
}
...
```

Considérations supplémentaires

Lorsque vous configurez l'option de substitution de propriété, les considérations suivantes s'appliquent :

- Par défaut, le propriétaire de l'objet source possède également le réplica. Amazon S3 réplique la version de l'objet et la liste ACL qui lui est associée.

  Si vous ajoutez la substitution de propriétaire, Amazon S3 réplique uniquement la version de l'objet, mais pas la liste ACL. En outre, Amazon S3 ne réplique pas les modifications ultérieures de la liste ACL de l'objet source. Amazon S3 définit la liste ACL sur le réplica qui accorde un contrôle total au propriétaire du compartiment de destination.
- Lorsque vous mettez à jour une configuration de réplication pour activer ou désactiver la substitution de propriétaire, les événements suivants se produisent.

- Si vous ajoutez l'option de substitution du propriétaire à la configuration de réplication :

  Quand Amazon S3 réplique une version de l'objet, il supprime la liste ACL associée à l'objet source. Il définit à la place la liste de contrôle d'accès (ACL) sur le réplica et accorde un contrôle total au propriétaire du compartiment de destination. Il ne réplique pas les modifications ultérieures de la liste de contrôle d'accès (ACL) de l'objet source. Toutefois, cette modification de liste de contrôle d'accès (ACL) ne s'applique pas aux versions d'objet répliquées avant la définition de l'option de substitution du propriétaire. Les mises à jour de liste de contrôle d'accès (ACL) pour les objets source répliqués avant la spécification de la substitution du propriétaire continuent donc d'être répliquées (car l'objet et ses réplicas continuent d'avoir le même propriétaire).
- Si vous supprimez l'option de substitution du propriétaire de la configuration de réplication :

  Amazon S3 réplique les nouveaux objets qui apparaissent dans le compartiment source et les listes ACL associées dans les compartiments de destination. Pour les objets répliqués avant la suppression de la substitution de propriétaire, Amazon S3 ne réplique pas les listes ACL car la modification de propriétaire des objets effectuée par Amazon S3 est toujours en vigueur. Les listes de contrôle d'accès (ACL) mises sur les versions d'objets qui étaient répliquées lorsque la substitution du propriétaire a été définie ne sont toujours pas répliquées.

Comment changer le propriétaire de la réplique

Lorsque les compartiments *`source`* et de *`destination`* d'une configuration de réplication appartiennent à des propriétaires différents Comptes AWS, vous pouvez demander à Amazon S3 de remplacer le propriétaire de la réplique par Compte AWS celui qui détient le compartiment de *`destination`*. Cet exemple explique comment utiliser la console Amazon S3 et comment AWS CLI modifier le propriétaire de la réplique. Pour plus d'informations, consultez Modification du propriétaire d'un réplica.

> **Note**
>
> Lorsque vous utilisez la réplication S3 et que les compartiments source et de destination appartiennent à des propriétaires différents Comptes AWS, le propriétaire du compartiment de destination peut désactiver les ACL (le propriétaire du compartiment imposant le paramètre Object Ownership) afin de remplacer la propriété de la réplique par le Compte AWS propriétaire du compartiment de destination. Ce paramètre imite le

comportement de remplacement du propriétaire existant sans avoir besoin d'une autorisation `s3:ObjectOwnerOverrideToBucketOwner`. Cela signifie que tous les objets répliqués dans le compartiment de destination avec le paramètre bucket owner enforced (propriétaire du compartiment imposé) appartiennent au propriétaire du compartiment de destination. Pour en savoir plus sur la propriété des objets, veuillez consulter Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Pour plus d'informations sur la configuration de la réplication à l'aide du chiffrement côté serveur AWS Key Management Service dans des scénarios entre comptes, consultez. Octroi d'autorisations supplémentaires pour les scénarios à plusieurs comptes

Utilisation de la console S3

Pour step-by-step obtenir des instructions, voirConfiguration d'une réplication pour des compartiments source et destination appartenant au même compte. Cette rubrique fournit des instructions pour définir la configuration de réplication lorsque les buckets appartiennent à des entités identiques ou différentes Comptes AWS.

En utilisant le AWS CLI

Pour modifier la propriété des répliques à l'aide de AWS CLI, vous créez des compartiments, activez le contrôle de version sur les compartiments, créez un rôle IAM qui autorise Amazon S3 à répliquer des objets et ajoutez la configuration de réplication au compartiment source. Dans la configuration de réplication, vous chargez Amazon S3 de changer le propriétaire du réplica. Vous devez aussi tester la configuration.

Pour modifier la propriété de la réplique lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS ()AWS CLI

1. Dans cet exemple, vous créez les compartiments *`source`* et *`destination`* dans deux compartiments différents Comptes AWS. Configurez le AWS CLI avec deux profils nommés. Cet exemple utilise des profils respectivement nommés `acctA` et `acctB`. Pour plus d'informations sur la définition de profils d'informations d'identification, consultez Profils nommés dans le Guide de l'utilisateur AWS Command Line Interface .

> **Important**
>
> Les profils que vous utilisez dans cet exercice doivent disposer des autorisations nécessaires. Par exemple, dans la configuration de réplication, vous spécifiez le rôle IAM qu'Amazon S3 peut endosser. Vous ne pouvez effectuer cette tâche que si le profil que vous utilisez dispose de l'autorisation `iam:PassRole`. Si vous utilisez les informations d'identification d'un administrateur pour créer un profil nommé, vous pouvez exécuter toutes les tâches. Pour plus d'informations, consultez la section Accorder à un utilisateur l'autorisation de transmettre un rôle à un AWS service dans le guide de l'utilisateur IAM.

Vous devez vérifier que ces profils disposent des autorisations nécessaires. Par exemple, la configuration de réplication inclut un rôle IAM qu'Amazon S3 peut endosser. Le profil nommé que vous utilisez pour attacher cette configuration à un compartiment ne peut effectuer cette tâche que s'il possède l'autorisation `iam:PassRole`. Si vous spécifiez des informations d'identification d'administrateur lors de la création de ces profils nommés, ils disposeront de toutes les autorisations. Pour plus d'informations, consultez la section Accorder à un utilisateur l'autorisation de transmettre un rôle à un AWS service dans le guide de l'utilisateur IAM.

2. Créez le compartiment *source* et activez la gestion des versions. Cet exemple crée le compartiment *source* dans la Région USA Est (Virginie du Nord) (us-east-1).

```
aws s3api create-bucket \
--bucket source \
--region us-east-1 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket source \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Créez un compartiment de *destination* et activez la gestion des versions. Cet exemple crée le compartiment de *destination* dans la Région USA Ouest (Oregon) (us-west-2). Utilisez un profil de Compte AWS différent de celui utilisé pour le compartiment *source*.

```
aws s3api create-bucket \
--bucket destination \
```

```
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctB
```

```
aws s3api put-bucket-versioning \
--bucket destination \
--versioning-configuration Status=Enabled \
--profile acctB
```

4. Vous devez ajouter des autorisations à la stratégie de votre compartiment de *destination* pour autoriser la modification de propriété du réplica.

   a. Enregistrez la stratégie suivante dans *destination-bucket-policy*.json.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "destination_bucket_policy_sid",
      "Principal": {
        "AWS": "source-bucket-owner-account-id"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ObjectOwnerOverrideToBucketOwner",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::destination/*"
      ]
    }
  ]
}
```

   b. Placez la stratégie précédente dans le compartiment de *destination* :

```
aws s3api put-bucket-policy --region $ {destination_region} --
bucket $ {destination} --policy file://destination_bucket_policy.json
```

5. Créez un rôle IAM. Vous précisez ce rôle dans la configuration de réplication que vous ajouterez ultérieurement au compartiment *source*. Amazon S3 endosse ce rôle pour répliquer des objets en votre nom. Vous créez un rôle IAM en deux étapes.

   - Créez un rôle
   - Attachez une stratégie d'autorisation au rôle.

   a. Créez un rôle IAM.

      i. Copiez la stratégie d'approbation suivante et enregistrez-la dans un fichier nommé `s3-role-trust-policy.json` dans le répertoire actif sur votre ordinateur local. Cette stratégie octroie à Amazon S3 les autorisations pour endosser le rôle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Exécutez la AWS CLI commande suivante pour créer un rôle.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

   b. Attachez une stratégie d'autorisation au rôle.

      i. Copiez la politique d'autorisations suivante et enregistrez-la dans un fichier nommé `s3-role-perm-pol-changeowner.json` dans le répertoire actuel de votre ordinateur local. Cette stratégie accorde des autorisations pour diverses actions sur les compartiments et les objets Amazon S3. Au cours des étapes suivantes, vous allez créer un rôle IAM et y attacher cette stratégie.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::source/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::source"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ObjectOwnerOverrideToBucketOwner",
            "s3:ReplicateTags",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":"arn:aws:s3:::destination/*"
      }
   ]
}
```

ii. Pour créer une stratégie et l'attacher au rôle, exécutez la commande suivante :

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
```

```
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

6. Ajoutez une configuration de réplication à votre compartiment source.

   a. AWS CLI Nécessite de spécifier la configuration de réplication au format JSON. Enregistrez la configuration JSON suivante dans un fichier nommé `replication.json` dans le répertoire actif sur votre ordinateur local. Dans la configuration, l'ajout de `AccessControlTranslation` permet d'indiquer que la propriété du réplica a été modifiée.

```
{
   "Role":"IAM-role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
         },
         "Status":"Enabled",
         "Destination":{
            "Bucket":"arn:aws:s3:::destination",
            "Account":"destination-bucket-owner-account-id",
            "AccessControlTranslation":{
               "Owner":"Destination"
            }
         }
      }
   ]
}
```

   b. Modifiez la configuration JSON en renseignant l'ID de compte du propriétaire du compartiment de *`destination`* et *`IAM-role-ARN`*. Enregistrez les modifications.

   c. Pour ajouter la configuration de réplication au compartiment source, exécutez la commande suivante : Indiquez le nom du compartiment *`source`*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket source \
```

```
--profile acctA
```

7. Vérifiez le propriétaire du réplica dans la console Amazon S3.

   a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   b. Ajoutez des objets au compartiment *source*. Vérifiez que le compartiment de *destination* contient les répliques d'objets et que le propriétaire des répliques a été remplacé par Compte AWS celui qui détient le compartiment de *destination*.

Utilisation des AWS SDK

Pour obtenir un exemple de code illustrant l'ajout d'une configuration de réplication, veuillez consulter Utilisation des AWS SDK. Vous devez modifier la configuration de réplication en conséquence. Pour obtenir des informations conceptuelles,veuillez consulter Modification du propriétaire d'un réplica.

Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC)

Le contrôle du délai de réplication Amazon S3 vous aide à respecter les règles de conformité et les besoins métier en matière de réplication des données, et améliore la visibilité des délais de réplication Amazon S3. Le contrôle du délai de réplication S3 permet de répliquer la plupart des objets que vous chargez dans Amazon S3 en quelques secondes, et 99,99 % de ces objets en 15 minutes.

S3 RTC inclut par défaut les métriques de réplication S3 et les notifications d'événements Amazon S3, que vous pouvez utiliser pour surveiller le nombre total d'opérations d'API S3 en attente de réplication, la taille totale des objets en attente de réplication et le temps de réplication maximal. Vous pouvez activer les métriques de réplication indépendamment de S3 RTC. Pour plus d'informations, consultez Surveillance de l'avancement des métriques de réplication. En outre, le contrôle du délai de réplication S3 fournit des événements `OperationMissedThreshold` et `OperationReplicatedAfterThreshold` qui notifient le propriétaire du compartiment si la réplication d'objet dépasse ou réplique après le seuil de 15 minutes.

Avec S3 RTC, les événements Amazon S3 peuvent vous avertir dans les rares cas où les objets ne se répliquent pas dans les 15 minutes et lorsque ces objets se répliquent après le seuil de 15 minutes. Les événements Amazon S3 sont disponibles via Amazon SQS, Amazon SNS ou. AWS Lambda Pour plus d'informations, consultez the section called “Notifications d'événements Amazon S3”.

Rubriques

Contrôle du temps de réplication S3

Vous pouvez commencer à utiliser le contrôle du délai de réplication S3 avec une règle de réplication nouvelle ou existante. Vous pouvez choisir d'appliquer votre règle de réplication à un compartiment S3 entier ou à des objets Amazon S3 avec un préfixe ou une balise spécifique. Lorsque vous activez S3 RTC, les mesures de réplication sont également activées sur votre règle de réplication.

Si vous utilisez la dernière version de la configuration de réplication, en spécifiant l'élément `Filter` dans une règle de configuration de réplication, Simple Storage Service (Amazon S3) ne réplique pas le marqueur de suppression par défaut. Vous pouvez toutefois ajouter la réplication des marqueurs de suppression aux non-tag-based règles.

> **Note**
>
> Les métriques de réplication sont facturées au même tarif que les métriques CloudWatch personnalisées Amazon. Pour plus d'informations, consultez CloudWatchles tarifs Amazon.

Pour plus d'informations sur la création d'une règle avec S3 RTC, consultez Activation du contrôle du temps de réplication S3 (S3 RTC).

Métriques de réplication avec contrôle du délai de réplication S3

Les règles de réplication avec le contrôle du délai de réplication S3 activé publient des métriques de réplication. Avec les métriques de réplication, vous pouvez surveiller le nombre total d'opérations d'API S3 en attente de réplication, la taille totale des objets en attente de réplication, la durée maximale de réplication vers la région de destination et le nombre total d'opérations pour lesquelles la réplication a échoué. Vous pouvez ensuite surveiller chaque jeu de données que vous répliquez séparément.

Les mesures de réplication sont disponibles dans les 15 minutes suivant l'activation de S3 RTC. Les métriques de réplication sont disponibles via la console Amazon S3, l'API Amazon S3, AWS les SDK, the AWS Command Line Interface (AWS CLI) et Amazon CloudWatch. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.

Pour plus d'informations sur la recherche de métriques de réplication via la console Amazon S3, consultez Affichage des métriques de réplication avec la console Amazon S3.

Utilisation des notifications d'événements Amazon S3 pour suivre les objets de réplication

Vous pouvez suivre la durée de réplication des objets qui n'ont pas été répliqués dans les 15 minutes en surveillant les notifications d'événements spécifiques que publie la fonction de contrôle du délai de réplication S3. Ces événements sont publiés lorsqu'un objet éligible à la réplication à l'aide de S3 RTC ne s'est pas répliqué dans les 15 minutes, et lorsque cet objet se réplique après le seuil de 15 minutes.

Les événements de réplication sont disponibles dans les 15 minutes suivant l'activation du contrôle du délai de réplication S3. Les événements Amazon S3 sont disponibles via Amazon SQS, Amazon SNS ou. AWS Lambda Pour plus d'informations, consultez Notifications d'événements Amazon S3.

Bonnes pratiques et directives de contrôle du délai de réplication S3

Lorsque vous répliquez des données dans Amazon S3 avec le contrôle du délai de réplication S3, suivez les directives de ces bonnes pratiques pour optimiser les performances de réplication de vos charges de travail.

Rubriques

- Directives en matière de performances de réplication et de débit de demandes Amazon S3
- Estimation de vos débits de demandes de réplication
- Dépassement des limites de débit de transfert de données RTC S3
- AWS KMS taux de demandes de réplication d'objets chiffrés

Directives en matière de performances de réplication et de débit de demandes Amazon S3

Vos applications peuvent exécuter des milliers de transactions par seconde en termes de performances de demande lors du chargement et de la récupération du stockage depuis Amazon S3. Par exemple, une application peut atteindre au moins 3 500 demandes PUT/COPY/POST/DELETE et 5 500 demandes GET/HEAD par seconde et par préfixe dans un compartiment S3, y compris les demandes que la réplication S3 effectue en votre nom. Il n'existe aucune limite au nombre de

préfixes dans un compartiment. Vous pouvez augmenter vos performances de lecture et d'écriture en effectuant une mise en parallèle des lectures. Par exemple, si vous créez 10 préfixes dans un compartiment S3 pour paralléliser les lectures, vous pouvez adapter vos performances de lecture à 55 000 demandes de lecture par seconde.

Amazon S3 effectue automatiquement une mise à l'échelle en réponse à des débits de demandes soutenus supérieurs à ces directives, ou à des débits de demandes soutenus simultanés à des demandes LIST. Tandis qu'Amazon S3 optimise en interne le débit de nouvelles demandes, vous pouvez recevoir temporairement des réponses HTTP 503 jusqu'à ce que l'optimisation soit terminée. Cela peut se produire avec des augmentations des débits de demande par seconde, ou lorsque vous activez S3 RTC pour la première fois. Au cours de ces périodes, votre latence de réplication peut augmenter. Le contrat de niveau de service (SLA) du contrôle du délai de réplication S3 ne s'applique pas aux périodes où les directives en matière de performances Amazon S3 sur les demandes par seconde sont dépassées.

Le SLA S3 RTC ne s'applique pas non plus pendant les périodes où votre débit de transfert de données de réplication dépasse la limite par défaut de 1 Gbit/s. Si vous pensez que votre débit de transfert de réplication va dépasser 1 Gbit/s, vous pouvez contacter le Centre AWS Support ou utiliser les Service Quotas pour demander une augmentation de votre limite.

Estimation de vos débits de demandes de réplication

Votre débit de demandes total, y compris les demandes que la réplication Amazon S3 effectue en votre nom, doit être conforme aux directives en matière de débit de demandes Amazon S3 pour les compartiments source et de destination de réplication. Pour chaque objet répliqué, la réplication Amazon S3 effectue jusqu'à cinq demandes GET/HEAD et une demande PUT vers le compartiment source, ainsi qu'une demande PUT vers chaque compartiment de destination.

Par exemple, si vous envisagez de répliquer 100 objets par seconde, la réplication Amazon S3 peut effectuer 100 demandes PUT supplémentaires en votre nom pour un total de 200 demandes PUT par seconde vers le compartiment S3 source. La réplication Amazon S3 peut également effectuer jusqu'à 500 demandes GET/HEAD (5 demandes GET/HEAD pour chaque objet répliqué).

> **Note**
>
> Vous n'engagez des coûts que pour une seule demande PUT par objet répliqué. Pour plus d'informations, consultez les informations de tarification dans la FAQ sur Amazon S3 relative à la réplication.

Dépassement des limites de débit de transfert de données RTC S3

Si vous pensez que votre dévit de transfert de données S3 Replication Time Control va dépasser la limite par défaut de 1 Gbit/s, contactez le Centre AWS Support ou utilisez les Service Quotas pour demander une augmentation de votre limite.

AWS KMS taux de demandes de réplication d'objets chiffrés

Lorsque vous répliquez des objets chiffrés avec le chiffrement côté serveur (SSE-KMS) à l'aide de la réplication Amazon S3, des limites de demandes par AWS Key Management Service seconde s'AWS KMS appliquent. AWS KMS peut rejeter une demande par ailleurs valide car votre taux de demandes dépasse la limite du nombre de demandes par seconde. Lorsqu'une demande est limitée, AWS KMS renvoie une `ThrottlingException` erreur. La limite de débit de AWS KMS demandes s'applique aux demandes que vous effectuez directement et aux demandes effectuées par Amazon S3 Replication en votre nom.

Par exemple, si vous prévoyez de répliquer 1 000 objets par seconde, vous pouvez soustraire 2 000 demandes de votre limite de taux de AWS KMS demandes. Le taux de requêtes par seconde qui en résulte est disponible pour vos AWS KMS charges de travail, à l'exception de la réplication. Vous pouvez utiliser les statistiques des AWS KMS demandes sur Amazon CloudWatch pour surveiller le taux total de AWS KMS demandes sur votre Compte AWS.

Activation du contrôle du temps de réplication S3 (S3 RTC)

Le contrôle du délai de réplication Amazon S3 vous aide à respecter les règles de conformité et les besoins métier en matière de réplication des données, et améliore la visibilité des délais de réplication Amazon S3. Le contrôle du délai de réplication S3 permet de répliquer la plupart des objets que vous chargez dans Amazon S3 en quelques secondes, et 99,99 % de ces objets en 15 minutes.

Avec S3 RTC, vous pouvez surveiller le nombre total et la taille des objets en attente de réplication, ainsi que le temps de réplication maximal vers la Région de destination. Les métriques de réplication sont disponibles via le Guide de CloudWatch l'utilisateur Amazon AWS Management Consoleet Amazon. Pour plus d'informations, consultez the section called “Métriques de réplication S3 dans CloudWatch ”.

Utilisation de la console S3

Pour step-by-step obtenir des instructions, voirConfiguration d'une réplication pour des compartiments source et destination appartenant au même compte. Cette rubrique fournit des

instructions pour activer S3 RTC dans votre configuration de réplication lorsque les buckets appartiennent à des entités identiques ou différentes. Comptes AWS

En utilisant le AWS CLI

AWS CLI Pour répliquer des objets lorsque S3 RTC est activé, vous devez créer des compartiments, activer le contrôle de version sur les compartiments, créer un rôle IAM qui autorise Amazon S3 à répliquer des objets et ajouter la configuration de réplication au compartiment source. La configuration de réplication nécessite l'activation du contrôle du délai de réplication S3.

Pour répliquer avec S3 RTC activé (AWS CLI)

- L'exemple suivant définit `ReplicationTime` et `Metric`, et ajoute la configuration de réplication au compartiment source.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Disabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::destination",
                "Metrics": {
                    "Status": "Enabled",
                    "EventThreshold": {
                        "Minutes": 15
                    }
                },
                "ReplicationTime": {
                    "Status": "Enabled",
                    "Time": {
                        "Minutes": 15
                    }
                }
            },
            "Priority": 1
        }
    ],
```

```
    "Role": "IAM-Role-ARN"
}
```

> **Important**
>
> La seule valeur valide pour `Metrics:EventThreshold:Minutes` et `ReplicationTime:Time:Minutes` est 15.

Utilisation du AWS SDK pour Java

Voici un exemple Java permettant d'ajouter une configuration de réplication avec le contrôle du délai de réplication S3 (S3 RTC).

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.DeleteMarkerReplication;
import software.amazon.awssdk.services.s3.model.Destination;
import software.amazon.awssdk.services.s3.model.Metrics;
import software.amazon.awssdk.services.s3.model.MetricsStatus;
import software.amazon.awssdk.services.s3.model.PutBucketReplicationRequest;
import software.amazon.awssdk.services.s3.model.ReplicationConfiguration;
import software.amazon.awssdk.services.s3.model.ReplicationRule;
import software.amazon.awssdk.services.s3.model.ReplicationRuleFilter;
import software.amazon.awssdk.services.s3.model.ReplicationTime;
import software.amazon.awssdk.services.s3.model.ReplicationTimeStatus;
import software.amazon.awssdk.services.s3.model.ReplicationTimeValue;

public class Main {

  public static void main(String[] args) {
    S3Client s3 = S3Client.builder()
      .region(Region.US_EAST_1)
      .credentialsProvider(() -> AwsBasicCredentials.create(
          "AWS_ACCESS_KEY_ID",
          "AWS_SECRET_ACCESS_KEY")
      )
      .build();

    ReplicationConfiguration replicationConfig = ReplicationConfiguration
      .builder()
      .rules(
```

```
          ReplicationRule
            .builder()
            .status("Enabled")
            .priority(1)
            .deleteMarkerReplication(
                DeleteMarkerReplication
                    .builder()
                    .status("Disabled")
                    .build()
            )
            .destination(
                Destination
                    .builder()
                    .bucket("destination_bucket_arn")
                    .replicationTime(
                        ReplicationTime.builder().time(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            ReplicationTimeStatus.ENABLED
                        ).build()
                    )
                    .metrics(
                        Metrics.builder().eventThreshold(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            MetricsStatus.ENABLED
                        ).build()
                    )
                    .build()
            )
            .filter(
                ReplicationRuleFilter
                    .builder()
                    .prefix("testtest")
                    .build()
            )
        .build())
        .role("role_arn")
        .build();

    // Put replication configuration
    PutBucketReplicationRequest putBucketReplicationRequest =
PutBucketReplicationRequest
      .builder()
```

```
        .bucket("source_bucket")
        .replicationConfiguration(replicationConfig)
        .build();

    s3.putBucketReplication(putBucketReplicationRequest);
  }
}
```

Pour plus d'informations, consultez Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC).

Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS)

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Certaines considérations particulières doivent être prises en compte lorsque vous répliquez des objets qui ont été chiffrés à l'aide du chiffrement côté serveur. Amazon S3 prend en charge les types suivants de chiffrement côté serveur :

- Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
- Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS)
- Chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS)
- Chiffrement côté serveur avec clés fournies par le client (SSE-C)

Pour plus d'informations sur le chiffrement côté serveur, consultez the section called "Chiffrement côté serveur".

Cette rubrique explique les autorisations dont vous avez besoin pour demander à Amazon S3 de répliquer des objets qui ont été chiffrés à l'aide du chiffrement côté serveur. Cette rubrique fournit également des éléments de configuration supplémentaires que vous pouvez ajouter, ainsi que des exemples de politiques AWS Identity and Access Management (IAM) qui accordent les autorisations nécessaires pour répliquer des objets chiffrés.

Pour un exemple avec des step-by-step instructions, voirActivation de la réplication pour les objets chiffrés. Pour obtenir des informations sur la création d'une configuration de réplication, veuillez consulter Vue d'ensemble de la réplication d'objets.

> **Note**
>
> Vous pouvez utiliser plusieurs régions AWS KMS keys dans Amazon S3. Cependant, Amazon S3 traite actuellement les clés multi-régions comme s'il s'agissait de clés à région unique et n'utilise pas les fonctions multi-régions de la clé. Pour en savoir plus, consultez la section Utilisation des clés multi-régions dans le Guide du développeur AWS Key Management Service .

Rubriques

- Comment le chiffrement par défaut du compartiment a un impact sur la réplication
- Réplication d'objets chiffrés avec SSE-C
- Réplication d'objets chiffrés avec SSE-S3, SSE-KMS ou DSSE-KMS
- Activation de la réplication pour les objets chiffrés

Comment le chiffrement par défaut du compartiment a un impact sur la réplication

Après avoir activé le chiffrement par défaut pour un compartiment de destination de réplication, le comportement de chiffrement suivant s'applique :

- Si des objets du compartiment source ne sont pas chiffrés, les objets réplica du compartiment de destination sont chiffrés à l'aide des paramètres de chiffrement par défaut du compartiment de destination. Par conséquent, les balises d'entité (ETags) des objets sources diffèrent des ETags des objets réplica. Si certaines de vos applications utilisent des ETags, vous devez les mettre à jour pour tenir compte de cette différence.
- Si les objets du compartiment source sont chiffrés à l'aide d'un chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3), d'un chiffrement côté serveur avec des clés AWS Key

Management Service (AWS KMS) (SSE-KMS) ou d'un chiffrement double couche côté serveur avec AWS KMS clés (DSSE-KMS), les objets répliques du compartiment de destination utilisent le même type de chiffrement que les objets source. Les paramètres de chiffrement par défaut du compartiment de destination ne sont pas utilisés.

Réplication d'objets chiffrés avec SSE-C

En utilisant le chiffrement côté serveur avec des clés fournies par le client (SSE-C), vous pouvez gérer vos propres clés de chiffrement propriétaires. Avec le SSE-C, vous gérez les clés tandis qu'Amazon S3 gère le processus de chiffrement et de déchiffrement. Vous devez fournir une clé de chiffrement dans le cadre de votre demande, mais vous n'avez pas besoin d'écrire de code pour effectuer le chiffrement ou le déchiffrement d'objets. Lorsque vous chargez un objet, Amazon S3 chiffre l'objet au moyen de la clé que vous avez fournie. Amazon S3 élimine ensuite cette clé de la mémoire. Lorsque vous récupérez un objet, vous devez fournir la même clé de chiffrement dans la demande. Pour plus d'informations, consultez the section called “Clés de chiffrement fournies par le client (SSE-C)”.

S3 Replication prend en charge les objets chiffrés avec SSE-C. Vous pouvez configurer la réplication d'objets SSE-C dans la console Amazon S3 ou à l'aide des AWS SDK, de la même manière que vous configurez la réplication pour les objets non chiffrés. Il n'existe pas d'autorisations SSE-C supplémentaires au-delà de celles actuellement requises pour la réplication.

La réplication S3 réplique automatiquement les objets chiffrés par SSE-C nouvellement chargés s'ils sont éligibles, conformément à votre configuration de réplication S3. Pour répliquer des objets existants dans vos compartiments, utilisez la réplication par lot S3. Pour plus d'informations sur la réplication d'objets existants, consultez the section called “Configuration de la réplication en direct” et the section called “Réplication d'objets existants”.

La réplication d'objets SSE-C n'entraîne pas de frais supplémentaires. Pour en savoir plus sur la tarification de la réplication, consultez la page de tarification d'Amazon S3.

Réplication d'objets chiffrés avec SSE-S3, SSE-KMS ou DSSE-KMS

Par défaut, Amazon S3 ne réplique pas les objets chiffrés avec SSE-KMS ou DSSE-KMS. Cette section explique les éléments de configuration supplémentaire que vous pouvez ajouter de manière à indiquer à Amazon S3 de répliquer ces objets.

Pour un exemple avec des step-by-step instructions, voirActivation de la réplication pour les objets chiffrés. Pour obtenir des informations sur la création d'une configuration de réplication, veuillez consulter Vue d'ensemble de la réplication d'objets.

Spécification d'informations supplémentaires dans la configuration de la réplication

Dans la configuration de réplication, procédez comme suit :

- Dans l'`Destination`élément de votre configuration de réplication, ajoutez l'ID de la clé symétrique gérée par le AWS KMS client que vous souhaitez qu'Amazon S3 utilise pour chiffrer les répliques d'objets, comme illustré dans l'exemple de configuration de réplication suivant.
- Validez explicitement votre choix en activant la réplication d'objets chiffrés avec des clés KMS (SSE-KMS ou DSSE-KMS). Pour valider, ajoutez l'élément `SourceSelectionCriteria`, comme illustré dans l'exemple de configuration de réplication suivant.

```
<ReplicationConfiguration>
   <Rule>
      ...
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
           <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
      </SourceSelectionCriteria>

      <Destination>
          ...
          <EncryptionConfiguration>
             <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same
 Région AWS as the destination bucket.</ReplicaKmsKeyID>
          </EncryptionConfiguration>
       </Destination>
      ...
   </Rule>
</ReplicationConfiguration>
```

> **Important**
>
> La clé KMS doit avoir été créée au même endroit Région AWS que les compartiments de destination.

La clé KMS doit être valide. L'opération d'API `PutBucketReplication` ne vérifie pas la validité des clés KMS. Si vous utilisez une clé KMS non valide, vous recevez le code de statut HTTP `200 OK` en réponse, mais la réplication échoue.

L'exemple suivant montre une configuration de réplication qui inclut des éléments de configuration facultatifs. Cette configuration de réplication a une règle. Cette règle s'applique aux objets dotés du préfixe de clé `Tax`. Amazon S3 utilise l'ID AWS KMS key spécifié pour chiffrer ces réplicas d'objets.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::example-s3-destination-bucket</Bucket>
         <EncryptionConfiguration>
            <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same
 Région AWS as the destination bucket. (S3 uses this key to encrypt object replicas.)</
ReplicaKmsKeyID>
         </EncryptionConfiguration>
      </Destination>
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
            <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
      </SourceSelectionCriteria>
   </Rule>
</ReplicationConfiguration>
```

Octroi d'autorisations supplémentaires pour le rôle IAM

Pour répliquer des objets chiffrés au repos à l'aide de SSE-S3, SSE-KMS ou DSSE-KMS, accordez les autorisations supplémentaires suivantes au rôle AWS Identity and Access Management (IAM) que vous spécifiez dans la configuration de réplication. Vous octroyez ces autorisations en mettant à jour la stratégie d'autorisations associée au rôle IAM.

- Action **`s3:GetObjectVersionForReplication`** pour les objets sources – Cette action permet à Amazon S3 de répliquer les objets non chiffrés et les objets créés avec un chiffrement côté serveur en utilisant une clé SSE-S3, SSE-KMS ou DSSE-KMS.

  > **Note**
  >
  > Nous vous recommandons d'utiliser l'action `s3:GetObjectVersionForReplication` à la place de l'action `s3:GetObjectVersion`, car `s3:GetObjectVersionForReplication` fournit uniquement à Amazon S3 le minimum d'autorisations nécessaires pour la réplication. De plus, l'action `s3:GetObjectVersion` permet la réplication des objets non chiffrés et des objets chiffrés avec SSE-S3, mais pas la réplication des objets chiffrés avec des clés KMS (SSE-KMS ou DSSE-KMS).

- **`kms:Decrypt`**et **`kms:Encrypt`** AWS KMS actions pour les clés KMS
  - Vous devez accorder des autorisations `kms:Decrypt` pour l' AWS KMS key utilisée pour déchiffrer l'objet source.
  - Vous devez accorder des autorisations `kms:Encrypt` pour la AWS KMS key utilisée pour chiffrer le réplica source.
- Action **`kms:GenerateDataKey`** pour la réplication d'objets en texte brut – Si vous répliquez des objets en texte brut dans un compartiment avec le chiffrement SSE-KMS ou DSSE-KMS activé par défaut, vous devez inclure l'autorisation `kms:GenerateDataKey` pour le contexte de chiffrement de destination et la clé KMS dans la politique IAM.

Nous vous recommandons de limiter ces autorisations uniquement aux compartiments et objets de destination en utilisant des clés de AWS KMS condition. Le Compte AWS titulaire du rôle IAM doit disposer d'autorisations `kms:Encrypt` et d'`kms:Decrypt`actions pour les clés KMS répertoriées dans la politique. Si les clés KMS appartiennent à une autre personne Compte AWS, le propriétaire des clés KMS doit accorder ces autorisations au Compte AWS titulaire du rôle IAM. Pour plus d'informations sur la gestion de l'accès à ces clés KMS, consultez la section Utilisation des politiques IAM AWS KMS dans le Guide du AWS Key Management Service développeur.

Clés de compartiment S3 et réplication

Pour utiliser la réplication avec une clé de compartiment S3, la AWS KMS key politique relative à la clé KMS utilisée pour chiffrer la réplique de l'objet doit inclure l'`kms:Decrypt`autorisation du principal appelant. L'appel à `kms:Decrypt` vérifie l'intégrité de la clé de compartiment S3 avant de l'utiliser. Pour plus d'informations, consultez Utilisation d'une clé de compartiment S3 avec réplication.

Lorsqu'une clé de compartiment S3 est activée pour le compartiment source ou de destination, le contexte de chiffrement est l'Amazon Resource Name (ARN) du compartiment et non l'ARN de l'objet (par exemple, `arn:aws:s3:::`*`bucket_ARN`*). Vous devez mettre à jour vos politiques IAM pour utiliser l'ARN du compartiment pour le contexte de chiffrement :

```
"kms:EncryptionContext:aws:s3:arn": [
"arn:aws:s3:::bucket_ARN"
]
```

Pour plus d'informations, consultez Contexte de chiffrement (`x-amz-server-side-encryption-context`) (dans la section « Utilisation de l'API REST ») et Modifications à prendre en compte avant d'activer une clé de compartiment S3.

Exemples de politiques : Utilisation du SSE-S3 et du SSE-KMS avec la réplication

L'exemple suivant de politiques IAM montre des instructions pour l'utilisation du SSE-S3 et du SSE-KMS avec la réplication.

Example : utilisation de SSE-KMS avec des compartiments de destination distincts

L'exemple suivant de politique présente des instructions pour l'utilisation du SSE-KMS avec des compartiments de destination distincts.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action": ["kms:Decrypt"],
         "Effect": "Allow",
         "Condition": {
             "StringLike": {
                  "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
                  "kms:EncryptionContext:aws:s3:arn": [
                      "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
                  ]
```

```
            }
         },
         "Resource": [
            "List of AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::example-s3-destination-bucket1/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same Région AWS as destination bucket 1). Used to
 encrypt object replicas created in destination bucket 1."
         ]
      },
      {
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::example-s3-destination-bucket2/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same Région AWS as destination bucket 2). Used to
 encrypt object replicas created in destination bucket 2."
         ]
      }
   ]
}
```

Example : réplication d'objets créés avec SSE-S3 et SSE-KMS

Vous trouverez ci-dessous une politique IAM complète qui accorde les autorisations nécessaires pour répliquer des objets non chiffrés, des objets créés avec le SSE-S3 et des objets créés avec le SSE-KMS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::example-s3-destination-bucket/key-prefix1*"
      },
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
```

```
            "StringLike":{
               "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
               ]
            }
         },
         "Resource":[
           "List of the AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-destination-bucket/prefix1*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same Région AWS as the destination bucket) to use
 for encrypting object replicas"
         ]
      }
   ]
}
```

Example : réplication d'objets avec des clés de compartiment S3

Voici une politique IAM complète qui accorde les autorisations nécessaires pour répliquer des objets avec des clés de compartiment S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
```

```
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::example-s3-destination-bucket/key-prefix1*"
      },
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-source-bucket"
               ]
            }
         },
         "Resource":[
           "List of the AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action":[
```

```
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-destination-bucket"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same Région AWS as the destination bucket) to use
 for encrypting object replicas"
         ]
      }
   ]
}
```

Octroi d'autorisations supplémentaires pour les scénarios à plusieurs comptes

Dans un scénario entre comptes, où les compartiments source et destination sont détenus par des entités différentes Comptes AWS, vous pouvez utiliser une clé KMS pour chiffrer les répliques d'objets. Le propriétaire de la clé KMS doit accorder au propriétaire du compartiment source l'autorisation d'utiliser la clé KMS.

> **Note**
>
> Si vous devez répliquer des données SSE-KMS entre comptes, votre règle de réplication doit spécifier une clé gérée par le client AWS KMS pour le compte de destination. Clés gérées par AWSn'autorisent pas l'utilisation entre comptes et ne peuvent donc pas être utilisés pour effectuer une réplication entre comptes.

Pour accorder au propriétaire du compartiment source l'autorisation d'utiliser la clé KMS (console AWS KMS)

1. Connectez-vous à la AWS KMS console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/kms.
2. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.

3. Pour afficher les clés de votre compte que vous créez et gérez vous-même, dans le volet de navigation, choisissez Clés gérées par le client.
4. Sélectionnez la clé KMS.
5. Sous la section Configuration générale, choisissez l'onglet Stratégie de clé.
6. Faites défiler la page vers le bas jusqu'à Autre Comptes AWS.
7. Choisissez Ajouter un autre Comptes AWS.

   La boîte Comptes AWS de dialogue Autre apparaît.
8. Dans la boîte de dialogue, choisissez Ajouter un autre Compte AWS. Pour arn:aws:iam::, saisissez l'ID de compte du compartiment source.
9. Sélectionnez Enregistrer les modifications.

Pour octroyer au propriétaire du compartiment source l'autorisation d'utiliser la clé KMS (AWS CLI)

- Pour plus d'informations sur la commande `put-key-policy` AWS Command Line Interface (AWS CLI), reportez-vous put-key-policyà la référence des AWS CLI commandes. Pour plus d'informations sur l'opération d'API `PutKeyPolicy` sous-jacente, consultez PutKeyPolicy dans la Référence d'API AWS Key Management Service.

AWS KMS considérations relatives aux quotas de transactions

Lorsque vous ajoutez de nombreux nouveaux objets AWS KMS chiffrés après avoir activé la réplication entre régions (CRR), vous pouvez rencontrer un ralentissement (erreurs HTTP). `503 Service Unavailable` La limitation survient lorsque le nombre de transactions AWS KMS par seconde dépasse le quota actuel. Pour plus d'informations, consultez Quotas dans le Guide du développeur AWS Key Management Service .

Pour demander l'augmentation d'un quota, utilisez Service Quotas. Pour plus d'informations, consultez Demande d'augmentation de quota. Si le Service Quotas n'est pas pris en charge dans votre région, ouvrez un AWS Support dossier.

Activation de la réplication pour les objets chiffrés

Par défaut, Amazon S3 ne réplique pas les objets chiffrés à l'aide d'un chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou d'un chiffrement double couche côté serveur avec clés (DSSE-KMS). AWS KMS Pour répliquer des objets chiffrés avec SSE-KMS ou DSS-KMS, vous devez modifier la configuration de réplication des compartiments de

façon à demander à Amazon S3 de répliquer ces objets. Cet exemple explique comment utiliser la console Amazon S3 et le AWS Command Line Interface (AWS CLI) pour modifier la configuration de réplication du compartiment afin de permettre la réplication d'objets chiffrés.

Pour plus d'informations, consultez Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

> **Note**
>
> Lorsqu'une clé de compartiment S3 est activée pour le compartiment source ou de destination, le contexte de chiffrement est l'Amazon Resource Name (ARN) du compartiment et non l'ARN de l'objet. Vous devez mettre à jour vos politiques IAM pour utiliser l'ARN du compartiment pour le contexte de chiffrement. Pour plus d'informations, consultez Clés de compartiment S3 et réplication.

> **Note**
>
> Vous pouvez utiliser plusieurs régions AWS KMS keys dans Amazon S3. Cependant, Amazon S3 traite actuellement les clés multi-régions comme s'il s'agissait de clés à région unique et n'utilise pas les fonctions multi-régions de la clé. Pour en savoir plus, consultez la section Utilisation des clés multi-régions dans le Guide du développeur AWS Key Management Service .

Utilisation de la console S3

Pour step-by-step obtenir des instructions, voirConfiguration d'une réplication pour des compartiments source et destination appartenant au même compte. Cette rubrique fournit des instructions pour définir une configuration de réplication lorsque les buckets appartiennent à des entités identiques ou différentes Comptes AWS.

En utilisant le AWS CLI

Pour répliquer des objets chiffrés avec le AWS CLI, procédez comme suit :

- Créez les compartiments source et de destination et activez la gestion des versions pour ces mêmes compartiments.

- Créez un rôle de service AWS Identity and Access Management (IAM) qui autorise Amazon S3 à répliquer des objets. Les autorisations accordées au rôle IAM incluent les autorisations nécessaires à la réplication des objets chiffrés.
- Ajoutez une configuration de réplication au compartiment source. La configuration de réplication fournit des informations sur la réplication des objets chiffrés à l'aide de clés KMS.
- Ajoutez des objets chiffrés au compartiment source.
- Testez la configuration pour vérifier que vos objets chiffrés sont répliqués dans le compartiment de destination.

Les procédures suivantes vous guident tout au long de ce processus.

Pour répliquer des objets chiffrés côté serveur (AWS CLI)

1. Dans cet exemple, vous allez créer les compartiments *example-s3-source-bucket* et *example-s3-destination-bucket* dans un même Compte AWS. Vous allez également définir un profil d'informations d'identification pour l' AWS CLI. Cet exemple utilise le nom de profil *acctA*.

   Pour plus d'informations sur la définition des profils d'identification, consultez la section Profils nommés dans le guide de l' AWS Command Line Interface utilisateur. Pour utiliser les commandes de cet exemple, remplacez *user input placeholders* par vos propres informations.
2. Utilisez les commandes suivantes pour créer le compartiment *DOC-EXAMPLE-SOURCE-BUCKET* et activer la gestion des versions sur celui-ci. Les commandes de l'exemple suivant créent le compartiment *DOC-EXAMPLE-SOURCE-BUCKET* dans la région USA Est (Virginie du Nord) (`us-east-1`).

   ```
   aws s3api create-bucket \
   --bucket DOC-EXAMPLE-SOURCE-BUCKET \
   --region us-east-1 \
   --profile acctA
   ```

   ```
   aws s3api put-bucket-versioning \
   --bucket DOC-EXAMPLE-SOURCE-BUCKET \
   --versioning-configuration Status=Enabled \
   --profile acctA
   ```

3. Utilisez les commandes suivantes pour créer le compartiment *DOC-EXAMPLE-DESTINATION-BUCKET* et activer la gestion des versions sur celui-ci. Les commandes de l'exemple suivant créent le compartiment *DOC-EXAMPLE-DESTINATION-BUCKET* dans la région USA Ouest (Oregon) (`us-west-2`).

   > **Note**
   >
   > Pour configurer une configuration de réplication lorsque les compartiments *DOC-EXAMPLE-SOURCE-BUCKET*et les *DOC-EXAMPLE-DESTINATION-BUCKET*compartiments se trouvent dans le même emplacement Compte AWS, vous utilisez le même profil. Dans cet exemple, nous utilisons *acctA*. Pour configurer la réplication lorsque les compartiments appartiennent à des Comptes AWS différents, spécifiez des profils différents pour chaque compte.

   ```
   aws s3api create-bucket \
   --bucket DOC-EXAMPLE-DESTINATION-BUCKET \
   --region us-west-2 \
   --create-bucket-configuration LocationConstraint=us-west-2 \
   --profile acctA
   ```

   ```
   aws s3api put-bucket-versioning \
   --bucket DOC-EXAMPLE-DESTINATION-BUCKET \
   --versioning-configuration Status=Enabled \
   --profile acctA
   ```

4. Ensuite, créez une fonction du service IAM. Vous allez spécifier ce rôle dans la configuration de réplication que vous ajouterez par la suite au compartiment *DOC-EXAMPLE-SOURCE-BUCKET*. Amazon S3 endosse ce rôle pour répliquer des objets en votre nom. Vous créez un rôle IAM en deux étapes.

   - Créez un rôle de service.
   - Attachez une stratégie d'autorisation au rôle.

   a. Pour créer une fonction du service IAM, procédez comme suit :

i. Copiez la stratégie d'approbation suivante et enregistrez-la dans un fichier nommé `s3-role-trust-policy-kmsobj.json` dans le répertoire actif sur votre ordinateur local. Cette stratégie accorde au principal de service Amazon S3 les autorisations nécessaires pour endosser le rôle et permettre ainsi à Amazon S3 d'effectuer des tâches en votre nom.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

ii. Utilisez la commande suivante pour créer le rôle :

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json  \
--profile acctA
```

b. Ensuite, attachez une stratégie d'autorisation au rôle. Cette stratégie accorde des autorisations pour diverses actions sur les compartiments et les objets Amazon S3.

i. Copiez la politique d'autorisations suivante et enregistrez-la dans un fichier nommé `s3-role-permissions-policykmsobj.json` dans le répertoire actuel de votre ordinateur local. Vous allez créer un rôle IAM et attacherez par la suite la stratégie à celui-ci.

> **Important**
>
> Dans la politique d'autorisation, vous spécifiez les identifiants de AWS KMS clé qui seront utilisés pour le chiffrement des *example-s3-destination-bucket* compartiments *example-s3-source-bucket* et. Vous devez créer deux clés KMS distinctes pour les compartiments *example-s3-source-*

bucket et *example-s3-destination-bucket*. AWS KMS keys ne sont pas partagés en dehors de Région AWS celui dans lequel ils ont été créés.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::example-s3-source-bucket",
            "arn:aws:s3:::example-s3-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLikeIfExists":{
               "s3:x-amz-server-side-encryption":[
                  "aws:kms",
                  "AES256",
                  "aws:kms:dsse"
               ],
               "s3:x-amz-server-side-encryption-aws-kms-key-id":[
                  "AWS KMS key IDs(in ARN format) to use for encrypting
 object replicas"
               ]
            }
         },
         "Resource":"arn:aws:s3:::example-s3-destination-bucket/*"
```

```
      },
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.us-east-1.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-source-bucket/*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key IDs(in ARN format) used to encrypt source
 objects."
         ]
      },
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.us-west-2.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::example-s3-destination-bucket/*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key IDs(in ARN format) to use for encrypting object
 replicas"
         ]
      }
   ]
}
```

ii. Créez une politique et attachez-la au rôle.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
```

```
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

5. Ensuite, ajoutez la configuration de réplication suivante au compartiment *example-s3-source-bucket*. Elle demande à Amazon S3 de répliquer les objets dotés du préfixe `Tax/` dans le compartiment *example-s3-destination-bucket*.

> **Important**
>
> Dans la configuration de réplication, spécifiez le rôle IAM qu'Amazon S3 peut endosser. Vous ne pouvez effectuer cette tâche que si vous disposez de l'autorisation `iam:PassRole`. Le profil que vous spécifiez dans la commande CLI doit disposer de cette autorisation. Pour plus d'informations, consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un Service AWS dans le Guide de l'utilisateur IAM.

```
 <ReplicationConfiguration>
  <Role>IAM-Role-ARN</Role>
  <Rule>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
       <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
      <Bucket>arn:aws:s3:::example-s3-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
```

```
</ReplicationConfiguration>
```

Pour ajouter une configuration de réplication au compartiment *example-s3-source-bucket*, procédez comme suit :

a. Vous AWS CLI devez spécifier la configuration de réplication au format JSON. Enregistrez la configuration JSON suivante dans un fichier (`replication.json`) dans le répertoire actuel sur votre ordinateur local.

```
{
   "Role":"IAM-Role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
            "Prefix":"Tax"
         },
         "Destination":{
            "Bucket":"arn:aws:s3:::example-s3-destination-bucket",
            "EncryptionConfiguration":{
               "ReplicaKmsKeyID":"AWS KMS key IDs (in ARN format) to use for
 encrypting object replicas"
            }
         },
         "SourceSelectionCriteria":{
            "SseKmsEncryptedObjects":{
               "Status":"Enabled"
            }
         }
      }
   ]
}
```

b. Modifiez le JSON pour indiquer les valeurs du compartiment *example-s3-destination-bucket*, des *AWS KMS key IDs (in ARN format)* et de l'*IAM-role-ARN*. Enregistrez les Modifications.

c. Utilisez la commande suivante pour ajouter la configuration de réplication à votre compartiment *example-s3-source-bucket*. Veillez à saisir le nom du compartiment *example-s3-source-bucket*.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket example-s3-source-bucket \
--profile acctA
```

6. Testez la configuration pour vérifier que les objets chiffrés ont été répliqués. Dans la console Amazon S3, procédez comme suit :

   a. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

   b. Dans le compartiment *example-s3-source-bucket*, créez un dossier nommé `Tax`.

   c. Ajoutez des exemples d'objet au dossier. Assurez-vous de choisir l'option de chiffrement et de spécifier votre clé KMS pour chiffrer les objets.

   d. Vérifiez que le compartiment *example-s3-destination-bucket* contient les réplicas d'objets et qu'ils ont été chiffrés avec la clé KMS que vous avez spécifiée dans la configuration. Pour plus d'informations, consultez the section called “Obtention du statut de réplication”.

Utilisation des AWS SDK

Pour obtenir un exemple de code montrant comment ajouter une configuration de réplication, consultez Utilisation des AWS SDK. Vous devez modifier la configuration de réplication en conséquence.

Pour obtenir des informations conceptuelles,veuillez consulter Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).

Répliquer les modifications de métadonnées avec la synchronisation des modifications de réplica Amazon S3

La synchronisation des modifications de réplica Amazon S3 peut vous aider à assurer la cohérences des métadonnées d'objet telles que les balises, les listes de contrôles d'accès (ACL) et les paramètres de verrouillage d'objet répliqués entre les réplicas et les objets source. Par défaut, Amazon S3 réplique les métadonnées des objets source vers les réplicas uniquement. Lorsque

la synchronisation des modifications de réplica est activée, Amazon S3 réplique les modifications apportées aux métadonnées apportées aux copies de réplica vers l'objet source, ce qui rend la réplication bidirectionnelle.

Activer de la synchronisation des modifications de réplica

Vous pouvez utiliser la synchronisation des modifications de réplica Amazon S3 avec des règles de réplication nouvelles ou existantes. Vous pouvez l'appliquer à un compartiment S3 entier ou à des objets Amazon S3 qui ont un préfixe spécifique.

Pour activer la synchronisation des modifications de réplica à l'aide de la console Amazon S3, consultez Exemples de configuration de la réplication en direct. Cette rubrique fournit des instructions pour activer la synchronisation des modifications des répliques dans votre configuration de réplication lorsque les buckets appartiennent à des entités identiques ou différentes Comptes AWS.

Pour activer la synchronisation des modifications des répliques à l'aide de AWS Command Line Interface (AWS CLI), vous devez ajouter une configuration de réplication au compartiment contenant les répliques `ReplicaModifications` activées. *`Pour configurer la réplication bidirectionnelle, créez une règle de réplication entre le compartiment source (example-s3-bucket1) et le compartiment contenant les répliques (example-s3-bucket2). Créez ensuite une deuxième règle de réplication entre le compartiment contenant les répliques (example-s3-bucket2) et le compartiment source (example-s3-bucket1).`* Les seaux peuvent être identiques ou différents. Régions AWS

> **Note**
>
> Vous devez activer la synchronisation des modifications de réplica sur les deux compartiments pour répliquer les modifications de métadonnées de réplica, telles que les listes de contrôle d'accès (ACL) des objets, les balises d'objet ou les paramètres de verrouillage d'objet sur les objets répliqués. Comme toutes les règles de réplication, ces règles peuvent être appliquées à l'ensemble du compartiment Amazon S3 ou à un sous-ensemble d'objets Amazon S3 filtrés par préfixe ou balises d'objet.

Dans l'exemple de configuration suivant, Amazon S3 réplique les modifications de métadonnées sous le préfixe *`Tax`* dans le bucket *`example-s3-bucket`*, qui contiendrait les objets source.

```
{
```

```
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "SourceSelectionCriteria": {
                "ReplicaModifications":{
                    "Status": "Enabled"
                }
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Pour obtenir des instructions complètes sur la création de règles de réplication à l'aide du AWS CLI, voirConfiguration d'une réplication pour des compartiments source et destination appartenant au même compte.

Répliquer des marqueurs de suppression entre les compartiments

Par défaut, lorsque la réplication S3 est activée et qu'un objet est supprimé dans le compartiment source, Amazon S3 ajoute un marqueur de suppression dans le compartiment source uniquement. Les données sont ainsi protégées contre les suppressions malencontreuses.

Si la réplication des marqueurs de suppression est activée, ces marqueurs sont copiés dans les compartiments de destination et Amazon S3 se comporte comme si l'objet avait été supprimé dans les compartiments source et de destination. Pour plus d'informations sur le fonctionnement des marqueurs de suppression, consultez Utilisation des marqueurs de suppression.

> **Note**
>
> La réplication des marqueurs de suppression n'est pas prise en charge pour les règles de réplication basées sur des balises. La réplication des marqueurs de suppression ne respecte pas non plus le SLA de 15 minutes convenu pour l'utilisation du contrôle de délai de réplication S3.

Si vous n'utilisez pas la dernière version de configuration de réplication, les opérations de suppression affecteront la réplication différemment. Pour plus d'informations, consultez Impact des opérations de suppression sur la réplication.

Activer la réplication des marqueurs de suppression

Vous pouvez commencer à utiliser la réplication des marqueurs de suppression avec une règle de réplication nouvelle ou existante. Vous pouvez l'appliquer à un compartiment S3 entier ou à des objets Amazon S3 qui ont un préfixe spécifique.

> **Note**
>
> Lorsque vous activez la réplication des marqueurs de suppression et que votre compartiment est doté d'une règle d'expiration du cycle de vie S3, les marqueurs de suppression ajoutés par la règle d'expiration du cycle de vie S3 ne seront pas répliqués dans le compartiment de destination.

Pour activer la réplication des marqueurs de suppression à l'aide de la console Amazon S3, consultez Utilisation de la console S3. Cette rubrique fournit des instructions pour activer la réplication des marqueurs de suppression dans votre configuration de réplication lorsque les buckets appartiennent à des entités identiques ou différentes Comptes AWS.

Pour activer la réplication des marqueurs de suppression à l'aide de AWS Command Line Interface (AWS CLI), vous devez ajouter une configuration de réplication au compartiment source en `DeleteMarkerReplication` activant.

Dans l'exemple de configuration suivant, les marqueurs de suppression sont répliqués dans le compartiment de destination *DOC-EXAMPLE-BUCKET* pour les objets sous le préfixe *Tax*.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Enabled"
            },
            "Destination": {
```

```
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Pour obtenir des instructions complètes sur la création de règles de réplication via le AWS CLI, reportez-vous Configuration d'une réplication pour des compartiments source et destination appartenant au même compte à la section Procédures de réplication.

Gérer ou suspendre la réplication en direct

La réplication dynamique est la copie automatique et asynchrone d'objets dans des compartiments identiques ou différents. Régions AWS Après avoir configuré votre configuration de réplication, Amazon S3 réplique les objets nouvellement créés et les mises à jour d'objets depuis un compartiment source vers un ou plusieurs compartiments de destination spécifiés.

Utilisez la console Amazon S3 pour ajouter des règles de réplication dans le compartiment source. Les règles de réplication définissent les objets du compartiment source à répliquer, ainsi que le compartiment de destination dans lequel les objets répliqués seront stockés. Pour plus d'informations sur la réplication, consultez Vue d'ensemble de la réplication d'objets.

Vous pouvez gérer les règles de réplication sur la page Réplication. Vous pouvez ajouter, afficher, activer, désactiver ou supprimer des règles de réplication. Vous pouvez également modifier la priorité de vos règles de réplication. Pour plus d'informations sur l'ajout de règles de réplication à un compartiment, consultez Utilisation de la console S3.

Pour gérer les règles de réplication d'un compartiment S3 à l'aide de la console Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans l'onglet Buckets à usage général, choisissez le nom du bucket que vous souhaitez.
4. Choisissez l'onglet Gestion, puis faites défiler la page vers le bas jusqu'à Règles de réplication.
5. Vous pouvez modifier vos règles de réplication de la manière suivante :
    - Pour activer ou désactiver une règle de réplication, cliquez sur le bouton d'option situé à gauche de la règle. Dans le menu Actions, choisissez Activer la règle ou Désactiver la règle.

Vous pouvez également désactiver, activer ou supprimer toutes les règles du bucket depuis le menu Actions.

> **Note**
>
> Si vous désactivez une règle de réplication puis que vous la réactivez ultérieurement, les objets nouveaux ou modifiés qui n'ont pas été répliqués lorsque la règle était désactivée ne sont pas automatiquement répliqués lors de la réactivation de la règle. Pour répliquer ces objets, vous devez utiliser S3 Batch Replication. Pour plus d'informations, consultez the section called “Réplication d'objets existants”.

- Pour modifier la priorité d'une règle, cliquez sur le bouton d'option situé à gauche de la règle, puis sélectionnez Modifier la règle.

  Vous devez définir les priorités des règles pour éviter les conflits provoqués par les objets inclus dans l'étendue de plusieurs règles. En cas de chevauchement de règles, Amazon S3 utilise la priorité des règles pour déterminer la règle à appliquer. Plus le nombre est élevé, plus la priorité est haute. Pour plus d'informations sur la priorité des règles, consultez Configuration de réplication.

Suspension ou arrêt de la réplication

Pour suspendre temporairement la réplication et la reprendre automatiquement ultérieurement, vous pouvez utiliser l'`aws:s3:bucket-pause-replication`action dans AWS Fault Injection Service. Pour plus d'informations, consultez aws:s3:bucket-pause-replicationet interrompez la réplication S3 dans le guide de AWS Fault Injection Service l'utilisateur.

Pour arrêter la réplication dans Amazon S3, nous vous recommandons de désactiver vos règles de réplication. Si vous désactivez une règle de réplication puis que vous la réactivez ultérieurement, les objets nouveaux ou modifiés qui n'ont pas été répliqués lorsque la règle était désactivée ne sont pas automatiquement répliqués lors de la réactivation de la règle. Pour répliquer ces objets, vous devez utiliser S3 Batch Replication. Pour plus d'informations, consultez the section called “Réplication d'objets existants”.

La réplication s'arrêtera également si vous supprimez le rôle AWS Identity and Access Management (IAM), les autorisations AWS Key Management Service (AWS KMS) ou les autorisations liées à la politique de compartiment qui accordent à Amazon S3 les autorisations requises. Toutefois, nous

ne recommandons pas ces approches car elles entraînent l'échec de la réplication. Amazon S3 signale le statut de réplication pour les objets affectés sous la forme `FAILED`. Si les autorisations sont restaurées ultérieurement, les objets marqués comme tels ne *`FAILED`* sont pas automatiquement répliqués. Pour répliquer ces objets, vous devez utiliser S3 Batch Replication.

Surveillance de la progression avec des métriques de réplication et des notifications d'événements S3

Les métriques de réplication S3 fournissent des indicateurs détaillés pour les règles de réplication dans votre configuration de réplication. Grâce aux métriques de réplication, vous pouvez suivre la minute-by-minute progression en suivant les octets en attente, les opérations en attente, les opérations dont la réplication a échoué et la latence de réplication.

Les métriques de réplication S3 sont activées automatiquement lorsque vous activez le contrôle du temps de réplication S3 (S3 RTC). Vous pouvez également activer les métriques de réplication S3 indépendamment du délai de réplication S3 lors de la création ou de la modification d'une règle. S3 RTC inclut d'autres fonctions telles qu'un contrat de niveau de service (SLA) et des notifications pour les seuils non respectés. Pour plus d'informations, consultez Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC).

Les métriques d'octets en attente, d'opérations en attente et de latence de réplication s'appliquent uniquement aux nouveaux objets répliqués avec la réplication interrégionale S3 (S3 CRR) ou la réplication S3 sur une même région (S3 SRR). La métrique d'échec de la réplication des opérations permet de suivre à la fois les nouveaux objets répliqués avec S3 CRR ou S3 SRR, et les objets existants qui sont répliqués avec la réplication par lot S3. Afin de vous aider à résoudre les problèmes de configuration, vous pouvez également configurer les notifications d'événement Amazon S3 pour recevoir les événements d'échec de la réplication.

Lorsqu'elles sont activées, les métriques de réplication S3 publient les métriques suivantes sur Amazon CloudWatch :

- Octets en attente de réplication : nombre total d'octets d'objets en attente de réplication pour une règle de réplication donnée.
- Latence de réplication : nombre maximal de secondes pendant lesquelles les compartiments de destination de réplication se trouvent derrière le compartiment source pour une règle de réplication donnée.

- Opérations en attente de réplication : nombre d'opérations en attente de réplication pour une règle de réplication donnée. Cette métrique suit les opérations liées aux objets, les marqueurs de suppression, les balises, les listes de contrôle d'accès (ACL) et le verrouillage d'objets S3.
- Opérations avec échec de la réplication : nombre d'opérations pour lesquelles la réplication a échoué pour une règle de réplication donnée. Cette métrique suit les opérations liées aux objets, les marqueurs de suppression, les balises, les listes de contrôle d'accès (ACL) et le verrouillage d'objets. Contrairement aux autres métriques de réplication, cette métrique s'applique aux nouveaux objets répliqués avec S3 CRR ou S3 SRR, et aux objets existants répliqués avec la réplication par lot S3.

> **Note**
>
> Opérations avec échec de la réplication suit les échecs de réplication S3 agrégés par minute. Pour identifier les objets spécifiques pour lesquels la réplication a échoué et les raisons de leur échec, abonnez-vous à l'événement `OperationFailedReplication` dans les notifications d'événements Amazon S3. Pour plus d'informations, consultez Recevoir des événements d'échec de réplication avec des notifications d'événements Amazon S3.

Si une tâche ne s'exécute pas du tout, les métriques ne sont pas envoyées à Amazon CloudWatch. Par exemple, votre tâche ne sera pas exécutée si vous ne disposez pas des autorisations nécessaires pour exécuter une tâche de réplication par lot S3, ou si les balises ou le préfixe de votre configuration de réplication ne correspondent pas.

Rubriques

- Activation des métriques de réplication S3
- Recevoir des événements d'échec de réplication avec des notifications d'événements Amazon S3
- Affichage des métriques de réplication avec S3 Storage Lens
- Affichage des métriques de réplication avec la console Amazon S3
- Raisons de l'échec de la réplication Amazon S3
- Obtention d'informations sur le statut de la réplication

Activation des métriques de réplication S3

Vous pouvez commencer à utiliser des métriques de réplication S3 avec une règle de réplication nouvelle ou existante. Vous pouvez choisir d'appliquer votre règle de réplication à un compartiment S3 entier ou à des objets Amazon S3 avec un préfixe ou une balise spécifique.

Cette rubrique fournit des instructions pour activer les métriques de réplication S3 dans votre configuration de réplication lorsque les compartiments source et de destination sont détenus par des Comptes AWS identiques ou différents.

Pour activer les métriques de réplication à l'aide de AWS Command Line Interface (AWS CLI), vous devez ajouter une configuration de réplication au compartiment source avec `Metrics` activé. Dans cet exemple de configuration, les objets sous le préfixe *`Tax`* sont répliqués dans le compartiment de destination *`DOC-EXAMPLE-BUCKET`* et des métriques sont générées pour ces objets.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "Metrics": {
                    "Status": "Enabled"
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Pour obtenir des instructions complètes sur la création de règles de réplication, consultez Configuration d'une réplication pour des compartiments source et destination appartenant au même compte.

Pour plus d'informations sur l'affichage des métriques de réplication dans la console S3, consultez Affichage des métriques de réplication avec la console Amazon S3.

> **Note**
>
> Les métriques de réplication S3 sont facturées au même tarif que les métriques CloudWatch personnalisées d'Amazon. Pour plus d'informations, consultez les CloudWatchtarifs Amazon.

Recevoir des événements d'échec de réplication avec des notifications d'événements Amazon S3

Les notifications d'événement S3 peuvent vous avertir dans les cas où les objets ne sont pas répliqués vers leur Région AWS de destination. Les événements Amazon S3 sont disponibles via Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) ou. AWS Lambda Pour plus d'informations, consultez the section called "Notifications d'événements Amazon S3".

Pour obtenir la liste des codes d'échec capturés par les notifications d'événements S3, consultez Raisons de l'échec de la réplication Amazon S3.

Affichage des métriques de réplication avec S3 Storage Lens

Pour obtenir des métriques détaillées sur la réplication S3, y compris des métriques sur le nombre de règles de réplication, vous pouvez utiliser Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Using S3 Storage Lens to protect your data (Utilisation de S3 Storage Lens pour protéger vos données). Pour obtenir la liste complète des métriques, consultez le glossaire des métriques de S3 Storage Lens.

Affichage des métriques de réplication avec la console Amazon S3

Il existe trois types de CloudWatch métriques Amazon pour Amazon S3 : les métriques de stockage, les métriques de demande et les métriques de réplication. Les métriques de réplication S3 sont activées automatiquement lorsque vous activez la réplication avec S3 Replication Time Control (S3 RTC) à l'aide de l'API Amazon S3 AWS Management Console ou de l'API Amazon S3. Vous pouvez également activer les métriques de réplication S3 indépendamment du délai de réplication S3 lors de la création ou de la modification d'une règle.

Les métriques de réplication suivent les ID de règle de la configuration de réplication. Un ID de règle de réplication peut être spécifique à un préfixe, à une balise ou à une combinaison des deux.

Pour plus d'informations sur CloudWatch les métriques pour Amazon S3, consultezSurveillance des métriques avec Amazon CloudWatch.

Prérequis

Activez une règle de réplication avec des métriques de réplication S3.

Pour afficher les métriques de réplication

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments. Dans la liste Compartiments, choisissez le nom du compartiment qui contient les objets pour lesquels vous souhaitez des métriques de réplication.
3. Choisissez l'onglet Métriques.
4. Sous Replication metrics (Métriques de réplication), choisissez Replication rules (Règles de réplication).
5. Choisissez Display charts (Afficher les graphiques).

   Amazon S3 affiche les graphiques Latence de réplication (en secondes), Octets en attente de réplication, Opérations en attente de réplication et Opérations avec échec de la réplication.

Vous pouvez ensuite afficher les métriques de réplication Latence de réplication (en secondes), Opérations en attente de réplication, Octets en attente de réplication) et Opérations avec échec de la réplication pour les règles que vous avez sélectionnées. Si vous utilisez S3 Replication Time Control, Amazon CloudWatch commence à communiquer les métriques de réplication 15 minutes après avoir activé S3 RTC sur la règle de réplication correspondante. Vous pouvez consulter les métriques de réplication sur la console Amazon S3 ou sur la CloudWatch console. Pour plus d'informations, consultez Métriques de réplication avec contrôle du délai de réplication S3.

> **Note**
>
> Vous pouvez également consulter les métriques détaillées relatives à la réplication S3 dans la console Amazon S3 à l'aide d'Amazon S3 Storage Lens. S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Pour plus d'informations, consultez Using S3 Storage Lens to protect your data (Utilisation de

S3 Storage Lens pour protéger vos données). Pour obtenir la liste complète des métriques, consultez le glossaire des métriques de S3 Storage Lens.

Raisons de l'échec de la réplication Amazon S3

Le tableau suivant répertorie les raisons des échecs de la réplication Amazon S3. Vous pouvez consulter ces raisons en recevant l'événement failureReason avec les notifications d'événements Amazon S3. Vous pouvez recevoir des notifications d'événements S3 via Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) ou. AWS Lambda Pour plus d'informations, consultez Notifications d'événements Amazon S3.

Vous pouvez également consulter ces raisons d'échec dans un rapport de fin de la réplication par lot S3. Pour plus d'informations, consultez Rapport de fin de la réplication par lot.

Raison de l'échec de la réplication	Description
`AssumeRoleNotPermitted`	Amazon S3 ne peut pas assumer le rôle AWS Identity and Access Management (IAM) spécifié dans la configuration de réplication ou dans le job Batch Operations.
`DstBucketInvalidRegion`	Le compartiment de destination ne se trouve pas dans le même compartiment Région AWS que celui spécifié par le job Batch Operations. Cette erreur est spécifique à la réplication par lot.
`DstBucketNotFound`	Amazon S3 n'est pas en mesure de trouver le compartiment de destination spécifié dans la configuration de la réplication.
`DstBucketObjectLockConfigMissing`	Pour répliquer des objets à partir d'un compartiment source avec le verrouillage d'objets activé, il doit également être activé dans le compartiment de destination. Cette erreur indique que le verrouillage d'objets n'est peut-être pas activé dans le compartim

Raison de l'échec de la réplication	Description
	ent de destination. Pour plus d'informations, consultez Considérations relatives au verrouillage d'objet.
`DstBucketUnversioned`	La gestion des versions n'est pas activée pour le compartiment de destination S3. Pour répliquer des objets avec la réplication S3, activez la gestion des versions pour le compartiment de destination.
`DstDelObjNotPermitted`	Amazon S3 n'est pas en mesure de répliquer des marqueurs de suppression vers le compartiment de destination. L'autorisation `s3:ReplicateDelete` peut être manquante pour le compartiment de destination.
`DstKmsKeyInvalidState`	L'état de la clé AWS Key Management Service (AWS KMS) du compartiment de destination n'est pas valide. Vérifiez et activez la AWS KMS clé requise. Pour plus d'informations sur la gestion des AWS KMS clés, consultez la section État des AWS KMS clés dans le manuel du AWS Key Management Service développeur.
`DstKmsKeyNotFound`	La AWS KMS clé configurée pour le compartiment de destination dans la configuration de réplication n'existe pas.
`DstMultipartCompleteNotPermitted`	Amazon S3 n'est pas en mesure de terminer les chargements partitionnés des objets dans le compartiment de destination. L'autorisation `s3:ReplicateObject` peut être manquante pour le compartiment de destination.

Raison de l'échec de la réplication	Description
`DstMultipartInitNotPermitted`	Amazon S3 n'est pas en mesure de lancer les chargements partitionnés des objets dans le compartiment de destination. L'autorisation `s3:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstMultipartPartUploadNotPermitted`	Amazon S3 n'est pas en mesure de charger les objets partitionnés dans le compartiment de destination. L'autorisation `s3:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstObjectHardDeleted`	La réplication par lot S3 ne prend pas en charge la réplication répétée d'objets supprimés avec l'ID de version de l'objet dans le compartiment de destination. Cette erreur est spécifique à la réplication par lot.
`DstPutAclNotPermitted`	Amazon S3 n'est pas en mesure de répliquer les listes de contrôle d'accès (ACL) de l'objet vers le compartiment de destination. L'autorisation `s3:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstPutLegalHoldNotPermitted`	Amazon S3 n'est pas en mesure d'effectuer une mise en suspens juridique du verrouillage d'objet lors de la réplication d'objets immuables. L'autorisation `s3:PutObjectLegalHold` peut être manquante pour le compartiment de destination. Pour plus d'informations, consultez Détentions légales.

Raison de l'échec de la réplication	Description
`DstPutObjectNotPermitted`	Amazon S3 n'est pas en mesure de répliquer des objets vers le compartiment de destination. Les autorisations `s3:ReplicateObject` ou `s3:ObjectOwnerOverrideToBucketOwner` peuvent être manquantes pour le compartiment de destination.
`DstPutTaggingNotPermitted`	Amazon S3 n'est pas en mesure de répliquer des balises d'objets vers le compartiment de destination. L'autorisation `s3:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstVersionNotFound`	Amazon S3 n'est pas en mesure de trouver la version d'objet requise dans le compartiment de destination pour lequel les métadonnées doivent être répliquées.
`InitiateReplicationNotPermitted`	Amazon S3 n'est pas en mesure de lancer la réplication sur des objets. L'autorisation `s3:InitiateReplication` peut être manquante pour la tâche d'opérations par lot. Cette erreur est spécifique à la réplication par lot.
`SrcBucketInvalidRegion`	Le compartiment source ne se trouve pas dans le même compartiment Région AWS que celui spécifié par le job Batch Operations. Cette erreur est spécifique à la réplication par lot.
`SrcBucketNotFound`	Amazon S3 n'est pas en mesure de trouver le compartiment source.

Raison de l'échec de la réplication	Description
`SrcBucketReplicationConfigMissing`	Amazon S3 n'a pas trouvé de configuration de réplication pour le compartiment source.
`SrcGetAclNotPermitted`	Amazon S3 n'est pas en mesure d'accéder à l'objet dans le compartiment source pour la réplication. L'autorisation `s3:GetObjectVersionAcl` peut être manquante pour l'objet du compartiment source. Les objets du compartiment source doivent appartenir au propriétaire du compartiment. Si les listes ACL sont activées, vérifiez si la propriété de l'objet est définie sur Propriétaire du compartiment préféré ou Créateur d'objet. Si la propriété de l'objet est définie sur Propriétaire du compartiment préféré, les objets du compartiment source doivent disposer de la liste ACL `bucket-owner-full-control` pour que le propriétaire du compartiment devienne propriétaire de l'objet. Le compte source peut prendre possession de tous les objets dans leur compartiment en définissant le paramètre Propriété d'objets sur Propriétaire du compartiment appliqué et en désactivant les listes ACL.
`SrcGetLegalHoldNotPermitted`	Amazon S3 n'est pas en mesure d'accéder aux informations de mise en suspens juridique du verrouillage des objets S3.
`SrcGetObjectNotPermitted`	Amazon S3 n'est pas en mesure d'accéder à l'objet dans le compartiment source pour la réplication. L'autorisation `s3:GetObjectVersionForReplication` peut être manquante pour le compartiment source.

Raison de l'échec de la réplication	Description
`SrcGetRetentionNotPermitted`	Amazon S3 n'est pas en mesure d'accéder aux informations de période de conservation du verrouillage des objets S3.
`SrcGetTaggingNotPermitted`	Amazon S3 n'est pas en mesure d'accéder aux informations de la balise d'objet depuis le compartiment source. L'autorisation `s3:GetObjectVersionTagging` peut être manquante pour le compartiment source.
`SrcHeadObjectNotPermitted`	Amazon S3 n'est pas en mesure de récupérer les métadonnées de l'objet depuis le compartiment source. L'autorisation `s3:GetObjectVersionForReplication` peut être manquante pour le compartiment source.
`SrcKeyNotFound`	Amazon S3 n'est pas en mesure de trouver la clé de l'objet source à répliquer. L'objet source a peut-être été supprimé avant la fin de la réplication.
`SrcKmsKeyInvalidState`	L'état de la AWS KMS clé du compartiment source n'est pas valide. Vérifiez et activez la AWS KMS clé requise. Pour plus d'informations sur la gestion des AWS KMS clés, consultez la section État des AWS KMS clés dans le manuel du AWS Key Management Service développeur.
`SrcObjectNotEligible`	Certains objets ne sont pas éligibles à la réplication. Cela peut être dû à la classe de stockage de l'objet ou aux balises de l'objet qui ne correspondent pas à la configuration de réplication.

Raison de l'échec de la réplication	Description
`SrcObjectNotFound`	L'objet source n'existe pas.
`SrcReplicationNotPending`	Amazon S3 a déjà répliqué cet objet. Cet objet n'est plus en attente de réplication.
`SrcVersionNotFound`	Amazon S3 n'est pas en mesure de trouver la version de l'objet source à répliquer. La version de l'objet source a peut-être été supprimée avant la fin de la réplication.

Rubriques en relation

Configuration des autorisations pour la réplication en direct

Résolution des problèmes de réplication

Obtention d'informations sur le statut de la réplication

Le statut de réplication peut vous aider à déterminer l'état actuel d'un objet répliqué. Le statut de réplication d'un objet source renvoie soit `PENDING`, `COMPLETED` ou `FAILED`. Le statut de réplication d'un réplica renvoie `REPLICA`.

Rubriques

- Vue d'ensemble des statuts de réplication
- Statut de la réplication en cas de réplication vers plusieurs compartiments de destination
- Statut de la réplication si la synchronisation des modifications de réplica Simple Storage Service (Amazon S3) est activée
- Recherche du statut de réplication

Vue d'ensemble des statuts de réplication

Dans le cadre de la réplication, vous disposez d'un compartiment source dans lequel vous configurez la réplication et d'un compartiment de destination dans lequel Amazon S3 réplique les objets. Lorsque vous demandez un objet (à l'aide de l'objet `GET`) ou des métadonnées d'objet (à l'aide de l'objet `HEAD`) à partir de ces compartiments, Amazon S3 renvoie l'en-tête `x-amz-replication-status` dans la réponse :

- Lorsque vous demandez un objet depuis le compartiment source, Amazon S3 renvoie l'en-tête `x-amz-replication-status` si l'objet demandé peut être répliqué.

  Par exemple, imaginons que vous spécifiez le préfixe d'objet `TaxDocs` dans votre configuration de réplication pour indiquer à Amazon S3 de ne répliquer que les objets dotés du préfixe de nom de clé `TaxDocs`. Tous les objets que vous chargez ayant ce préfixe de nom de clé (par exemple, `TaxDocs/document1.pdf`) seront répliqués. Pour les demandes d'objet avec ce préfixe de nom de clé, Amazon S3 renvoie l'en-tête `x-amz-replication-status` avec l'une des valeurs suivantes pour le statut de réplication de l'objet : `PENDING`, `COMPLETED` ou `FAILED`.

  > **Note**
  >
  > Si la réplication d'objet échoue après avoir chargé un objet, vous ne pouvez pas relancer la réplication. Vous devez recharger l'objet. Les objets passent à l'état `FAILED` en cas de problèmes, par exemple si les autorisations de rôle de réplication, les autorisations AWS KMS ou les autorisations de compartiment sont manquantes. Pour les échecs temporaires, par exemple si un compartiment ou une Région n'est pas disponible, le statut de réplication ne passera pas à `FAILED`, mais restera `PENDING`. Une fois la ressource remise en ligne, S3 reprendra la réplication de ces objets.

- Lorsque vous demandez un objet à partir du compartiment de destination, si l'objet demandé est un réplica créé par Amazon S3, Amazon S3 renvoie l'en-tête `x-amz-replication-status` avec la valeur `REPLICA`.

> **Note**
>
> Avant de supprimer un objet d'un compartiment source pour lequel la réplication a été activée, contrôlez le statut de réplication de l'objet pour vérifier qu'il a été répliqué.
> Si la configuration du cycle de vie est activée sur le compartiment source, Amazon S3 interrompt les actions de cycle de vie tant que les statuts des objets ne sont pas marqués comme `COMPLETED` ou `FAILED`.

Statut de la réplication en cas de réplication vers plusieurs compartiments de destination

Lorsque vous répliquez des objets vers plusieurs compartiments de destination, l'en-tête `x-amz-replication-status` agit différemment. L'en-tête de l'objet source renvoie uniquement la valeur

`COMPLETED` lorsque la réplication a réussi vers toutes les destinations. L'en-tête reste à la valeur `PENDING` jusqu'à ce que la réplication soit terminée pour toutes les destinations. Si la réplication échoue vers une ou plusieurs destinations, l'en-tête renvoie `FAILED`.

Statut de la réplication si la synchronisation des modifications de réplica Simple Storage Service (Amazon S3) est activée

Lorsque vos règles de réplication activent la synchronisation des modifications de réplica Simple Storage Service (Amazon S3), les statuts des réplicas peuvent être différents de `REPLICA`. Si des modifications de métadonnées sont en cours de réplication, l'en-tête `x-amz-replication-status` renvoie `PENDING`. Si la synchronisation des modifications du réplica ne parvient pas à répliquer les métadonnées, l'en-tête renvoie `FAILED`. Si les métadonnées sont répliquées correctement, les réplicas retournent l'en-tête `REPLICA`.

Recherche du statut de réplication

Pour obtenir le statut de réplication des objets dans un compartiment, utilisez l'outil d'inventaire Amazon S3. Amazon S3 envoie un fichier CSV au compartiment de destination que vous spécifiez dans la configuration d'inventaire. Vous pouvez également utiliser Amazon Athena pour interroger le statut de réplication dans le rapport d'inventaire. Pour plus d’informations sur l’inventaire Amazon S3, consultez Inventaire Simple Storage Service (Amazon S3).

Vous pouvez également connaître l'état de réplication de l'objet à l'aide de la console, du AWS Command Line Interface (AWS CLI) ou du AWS SDK.

Utilisation de la console S3

Dans la console S3, vous pouvez afficher le statut de réplication d'un objet sur la page Détails de l'objet sous la Vue d'ensemble de la gestion des objets.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, sélectionnez le nom de votre compartiment.
3. Dans la liste Objets , sélectionnez le nom de l'objet.
4. Sous l'onglet Properties (Propriétés), vous pouvez voir le Replication status (Statut de réplication) sous Object management overview (Présentation de la gestion des objets).

En utilisant le AWS CLI

Utilisez la commande `head-object` comme suit pour récupérer les métadonnées de l'objet.

```
aws s3api head-object --bucket source-bucket --key object-key --version-id object-version-id
```

La commande renvoie les métadonnées de l'objet, y compris `ReplicationStatus` comme illustré dans l'exemple de réponse suivant.

```
{
   "AcceptRanges":"bytes",
   "ContentType":"image/jpeg",
   "LastModified":"Mon, 23 Mar 2015 21:02:29 GMT",
   "ContentLength":3191,
   "ReplicationStatus":"COMPLETED",
   "VersionId":"jfnW.HIMOfYiD_9rGbSkmroXsFj3fqZ.",
   "ETag":"\"6805f2cfc46c0f04559748bb039d69ae\"",
   "Metadata":{

   }
}
```

Utilisation des AWS SDK

Les fragments de code suivants obtiennent l'état de réplication avec le AWS SDK for Java et AWS SDK for .NET, respectivement.

Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,
 key);
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);

System.out.println("Replication Status : " +
 metadata.getRawMetadataValue(Headers.OBJECT_REPLICATION_STATUS));
```

.NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest
    {
         BucketName = sourceBucket,
         Key        = objectKey
    };
```

```
GetObjectMetadataResponse getmetadataResponse =
 client.GetObjectMetadata(getmetadataRequest);
Console.WriteLine("Object replication status: {0}",
 getmetadataResponse.ReplicationStatus);
```

Réplication d'objets existants via la réplication par lot S3

Avec S3 Batch Replication, vous pouvez répliquer les types d'objets suivants :

- Objets qui existaient avant la mise en place d'une configuration de réplication
- Objets précédemment répliqués
- Objets dont la réplication a échoué

Vous pouvez répliquer ces objets à la demande à l'aide d'une tâche Batch Operations. La réplication par lots S3 est différente de la réplication en direct, qui réplique automatiquement et en continu les nouveaux objets dans les compartiments Amazon S3.

Pour commencer à utiliser Batch Replication, vous pouvez :

- Lancer la réplication par lots pour une nouvelle règle ou destination de réplication : vous pouvez créer une tâche de réplication par lots unique lorsque vous créez la première règle d'une nouvelle configuration de réplication ou lorsque vous ajoutez une nouvelle destination à une configuration existante via la console Amazon S3.
- Lancer la réplication par lots pour une configuration de réplication existante : vous pouvez créer une nouvelle tâche de réplication par lots en utilisant S3 Batch Operations via la console Amazon S3, le AWS Command Line Interface (AWS CLI), AWS les SDK ou l'API REST Amazon S3.

Lorsque la tâche de réplication par lot est terminée, vous recevez un rapport de fin d'opérations. Pour plus d'informations sur l'utilisation du rapport pour examiner la tâche, consultez Suivi de l'état de la tâche et des rapports de fin de tâche.

Considérations sur la réplication par lot S3

- Votre compartiment source doit être associé à une configuration de réplication existante. Pour activer la réplication, consultez la section Configuration de la réplication en direct et Exemples de configuration de la réplication en direct.

- Si S3 Lifecycle est configuré pour votre compartiment, nous vous recommandons de désactiver vos règles de cycle de vie lorsque la tâche de réplication par lots est active. Cela permet de garantir la parité entre les compartiments source et de destination. Dans le cas contraire, ces compartiments risquent de diverger et le compartiment de destination ne sera pas une réplique exacte du compartiment source. Par exemple, imaginez le scénario suivant:
  - Votre compartiment source contient plusieurs versions d'un objet et un marqueur de suppression sur cet objet.
  - Vos compartiments source et de destination disposent d'une configuration de cycle de vie pour retirer les marqueurs de suppression expirés.

  Dans ce scénario, Batch Replication peut répliquer le marqueur de suppression dans le compartiment de destination avant de répliquer les versions des objets. Ce comportement peut avoir pour conséquence que votre configuration du cycle de vie marque le marqueur de suppression comme expiré et que le marqueur de suppression soit supprimé du compartiment de destination avant que les versions des objets ne soient répliquées.
- Le rôle AWS Identity and Access Management (IAM) que vous spécifiez pour exécuter la tâche Batch Operations doit disposer des autorisations nécessaires pour effectuer l'opération de réplication par lots sous-jacente. Pour plus d'informations sur la création de rôles IAM, consultez Configuration des politiques IAM pour la réplication par lot.
- La réplication par lots nécessite un manifeste, qui peut être généré par Amazon S3. Le manifeste généré doit être stocké dans le même emplacement Région AWS que le compartiment source. Si vous choisissez de ne pas générer le manifeste, vous pouvez fournir un rapport d'inventaire Amazon S3 ou un fichier CSV contenant les objets que vous souhaitez répliquer.
- La réplication par lots ne prend pas en charge la réplication d'objets supprimés avec l'ID de version de l'objet depuis le compartiment de destination. Pour répéter la réplication de ces objets, vous pouvez copier les objets sources en place avec une tâche de copie par lot. La copie de ces objets sur place crée de nouvelles versions des objets dans le compartiment source et lance automatiquement la réplication vers le compartiment de destination. La suppression et la recréation du compartiment de destination n'initient pas la réplication.

  Pour plus d'informations sur Batch Copy, consultezExemples qui utilisent des opérations par lot pour copier des objets.
- Si vous utilisez une règle de réplication sur le compartiment S3, veillez à mettre à jour votre configuration de réplication en accordant au rôle IAM attaché à la règle de réplication les autorisations appropriées pour répliquer des objets. Ce rôle IAM doit disposer des autorisations nécessaires pour effectuer la réplication à la fois sur les compartiments source et de destination.

- Si vous soumettez plusieurs tâches de réplication par lots pour le même compartiment dans un court laps de temps, Amazon S3 exécutera ces tâches simultanément.
- Si vous soumettez plusieurs tâches de réplication par lots pour deux compartiments différents, sachez qu'Amazon S3 peut ne pas exécuter toutes les tâches simultanément. Si vous dépassez le nombre de tâches de réplication par lots pouvant être exécutées simultanément sur votre compte, Amazon S3 interrompra les tâches les moins prioritaires pour travailler sur les tâches les plus prioritaires. Une fois les tâches les plus prioritaires terminées, toutes les tâches en pause redeviennent actives.
- La réplication par lots n'est pas prise en charge pour les objets stockés dans les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.
- Pour répliquer par lots des objets S3 Intelligent-Tiering stockés dans les niveaux de stockage Archive Access ou Deep Archive Access, vous devez d'abord lancer une demande de restauration et attendre que les objets soient déplacés vers le niveau Frequent Access.

Spécification d'un manifeste pour une tâche de réplication par lot

Un manifeste est un objet Amazon S3 qui contient les clés d’objet sur lesquelles Amazon S3 doit agir. Si vous souhaitez créer une tâche de réplication par lots, vous devez fournir un manifeste généré par l'utilisateur ou demander à Amazon S3 de générer un manifeste en fonction de votre configuration de réplication.

Si vous fournissez un manifeste généré par l'utilisateur, il doit prendre la forme d'un rapport d'inventaire Amazon S3 ou d'un fichier CSV. Si les objets de votre manifeste sont dans un compartiment versionné, vous devez spécifier les ID de version des objets. Seul l'objet dont l'ID de version est spécifié dans le manifeste sera répliqué. Pour en savoir plus sur la spécification d'un manifeste, consultez Spécification d'un manifeste.

Si vous choisissez de demander à Amazon S3 de générer un fichier manifeste en votre nom, les objets répertoriés utiliseront le même compartiment source, le même préfixe et les mêmes balises que toutes vos configurations de réplication du compartiment source. Avec un manifeste généré, Amazon S3 répliquera toutes les versions éligibles de vos objets.

> **Note**
>
> Si vous choisissez qu'Amazon S3 génère le manifeste, celui-ci doit être stocké dans le même compartiment Région AWS que le compartiment source.

Filtres pour une tâche de réplication par lot

Lorsque vous créez votre tâche de réplication par lots, vous pouvez éventuellement spécifier des filtres supplémentaires, tels que la date de création de l'objet et l'état de réplication, afin de réduire l'étendue de la tâche.

Vous pouvez filtrer les objets à répliquer sur la base de la valeur `ObjectReplicationStatuses`, en fournissant une ou plusieurs des valeurs suivantes :

- `"NONE"` – Indique qu'Simple Storage Service (Amazon S3) n'a jamais tenté de répliquer l'objet auparavant.
- `"FAILED"`— Indique qu'Amazon S3 a déjà tenté, sans succès, de répliquer l'objet.
- `"COMPLETED"` – Indique qu'Simple Storage Service (Amazon S3) a déjà répliqué l'objet avec succès.
- `"REPLICA"`— Indique qu'il s'agit d'un objet répliqué qu'Amazon S3 a répliqué depuis une autre source.

Pour plus d'informations sur les statuts de la réplication, consultez Obtention d'informations sur le statut de la réplication.

Si vous ne filtrez pas votre tâche de réplication par lots, Batch Operations tentera de répliquer tous les objets (quels qu'ils soient`ObjectReplicationStatus`) de votre manifeste qui répondent aux règles de votre configuration de réplication, à l'exception de certains objets qui ne sont pas répliqués par défaut. Pour plus d’informations, consultez the section called “Qu'est-ce qui n'est pas répliqué avec les configurations de réplication ?”.

En fonction de votre objectif, vous pouvez `ObjectReplicationStatuses` définir une ou plusieurs des valeurs suivantes :

- Pour répliquer uniquement les objets existants qui n'ont jamais été répliqués, incluez uniquement. `"NONE"`
- Pour réessayer de répliquer uniquement les objets qui n'avaient pas pu être répliqués auparavant, incluez uniquement. `"FAILED"`
- Pour répliquer des objets existants et réessayer de répliquer des objets qui n'avaient pas pu être répliqués auparavant, incluez les deux et. `"NONE" "FAILED"`
- Pour remplir un compartiment de destination avec des objets répliqués vers une autre destination, incluez. `"COMPLETED"`

- Pour répliquer des objets précédemment répliqués, incluez. `"REPLICA"`

Rapport de fin de la réplication par lot

Lorsque vous créez une tâche de réplication par lot, vous pouvez demander un rapport de fin CSV. Ce rapport présente les objets, les codes de réussite ou d'échec de la réplication, les résultats et les descriptions. Pour plus d'informations sur le suivi des tâches et les rapports d'achèvement, consultezRapports de fin de tâche.

Pour obtenir la liste des codes d'échec de réplication et leur description, consultezRaisons de l'échec de la réplication Amazon S3.

Pour plus d'informations sur la résolution des problèmes liés à la réplication par lots, consultezErreurs de réplication par lot.

Démarrer avec la réplication par lot

Pour en savoir plus sur l'utilisation de la réplication par lot, consultez Tutoriel : Réplication d'objets existants dans vos compartiments Amazon S3 avec la réplication par lot S3.

Configuration des politiques IAM pour la réplication par lot

Dans la mesure où la réplication par lot S3 est un type de tâche d'opérations par lot, vous devez créer un rôle (IAM) d'opérations par lot AWS Identity and Access Management permettant d'accorder à Simple Storage Service (Amazon S3) les autorisations nécessaires pour effectuer des actions en votre nom. Vous devez également associer une politique IAM de réplication par lot au rôle IAM Opérations par lot. L'exemple suivant montre comment créer un rôle IAM qui donne aux opérations par lot l'autorisation de lancer une tâche de réplication par lot.

Créer un rôle et une politique IAM

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Sous Access Management (Gestion des accès), choisissez Roles (Rôles).
3. Choisissez Create Role (Créer un rôle).
4. Choisissez Service AWS comme type d'entité de confiance, Amazon S3 comme service, et S3 Batch Operations (Opérations par lot S3) comme cas d'utilisation.
5. Sélectionnez Suivant : autorisations.

6. Choisissez Create Policy (Créer une politique).
7. Choisissez JSON et insérez l'une des politiques suivantes en fonction de votre manifeste.

> **Note**
>
> Des autorisations différentes sont nécessaires selon que vous générez un manifeste ou que vous en fournissez un. Pour plus d'informations, veuillez consulter Spécification d'un manifeste pour une tâche de réplication par lot.

Politique d'utilisation et de stockage d'un manifeste généré par S3

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** manifest bucket ***/*"
```

```
            ]
        },
        {
            "Effect":"Allow",
            "Action":[
               "s3:PutObject"
            ],
            "Resource":[
               "arn:aws:s3:::*** completion report bucket ****/*",
               "arn:aws:s3:::*** manifest bucket ****/*"
            ]
        }
    ]
}
```

Politique en cas d'utilisation d'un manifeste fourni par l'utilisateur

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***/*"
         ]
      },
            {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** manifest bucket ***/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
```

```
            ],
            "Resource":[
               "arn:aws:s3:::*** completion report bucket ****/*"
            ]
         }
   ]
}
```

8. Choisissez Suivant : Balises.
9. Choisissez Next: Review (Suivant : Vérification).
10. Choisissez un nom pour la politique, puis Create policy (Créer une politique).
11. Attachez cette politique à votre rôle et choisissez Next: Tags (Suivant : Identifications).
12. Choisissez Suivant : vérification.
13. Choisissez un nom pour le rôle, puis Create role (Créer un rôle).

Vérifier la politique d'approbation

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Sous Access management (Gestion des accès), choisissez Roles (Rôles) et sélectionnez le rôle que vous venez de créer.
3. Sous l'onglet Trust relationships (Relations d'approbation), choisissez Edit trust relationship (Modifier la relation d'approbation).
4. Vérifiez que ce rôle utilise la politique d'approbation suivante :

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Créer une tâche de réplication par lot pour une première règle de réplication ou une nouvelle destination

Lorsque vous créez la première règle dans une nouvelle configuration de réplication ou que vous ajoutez une nouvelle destination à une configuration existante via le AWS Management Console, vous pouvez éventuellement créer une tâche de réplication par lots.

Pour utiliser la réplication par lot destinée à une configuration existante sans ajouter de nouvelle destination, consultez Créer une tâche de réplication par lot pour les règles de réplication existantes.

Utilisation de la réplication par lots pour une nouvelle règle de réplication ou une nouvelle destination via AWS Management Console

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient les objets que vous souhaitez répliquer.
3. Pour créer une nouvelle règle de réplication ou modifier une règle existante, choisissez Gestion et faites défiler l'écran jusqu'à Règles de réplication :

   - Pour créer une nouvelle règle de réplication, choisissez Créer une règle de réplication.

     > **Note**
     >
     > Pour obtenir des exemples de configuration d'une règle de réplication de base, consultez Exemples de configuration de la réplication en direct.

   - Pour modifier une règle de réplication existante, sélectionnez la règle, puis choisissez Modifier la règle.
4. Créez votre nouvelle règle de réplication ou modifiez la destination de votre règle de réplication existante, puis choisissez Enregistrer.

   Une fois que vous avez créé la première règle dans une nouvelle configuration de réplication ou modifié une configuration existante pour ajouter une nouvelle destination, la boîte de dialogue Répliquer des objets existants ? s'affiche et vous permet de créer une tâche de réplication par lots.
5. Si vous souhaitez exécuter cette tâche maintenant, choisissez Oui, répliquer les objets existants.

Si vous voulez exécuter cette tâche ultérieurement; choisissez Non, ne pas répliquer les objets existants.

6. Créez votre tâche de réplication par lots S3. La tâche de réplication par lot S3 comporte plusieurs paramètres :

   Option d'exécution de tâche

   Si vous souhaitez que la tâche de réplication par lots S3 soit exécutée immédiatement, vous pouvez choisir Exécution automatique de la tâche lorsqu'elle est prête. Si vous souhaitez exécuter la tâche ultérieurement, choisissez Exécution automatique de la tâche lorsqu'elle est prête.

   Si vous choisissez Job runs automatically when ready (La tâche s'exécute automatiquement une fois prête), vous ne pourrez pas créer ni enregistrer un manifeste d'opérations par lot. Pour enregistrer le manifeste des opérations par lot, choisissez Job waits to be run when ready (La tâche attend d'être exécutée lorsqu'elle est prête).

   Manifeste des opérations par lot

   Le manifeste est la liste de tous les objets sur lesquels vous souhaitez exécuter l'action spécifiée. Vous pouvez choisir d'enregistrer le manifeste des opérations par lot. Comme pour les fichiers Inventaire S3, le manifeste est enregistré en tant que fichier CSV et stocké dans un compartiment. Pour en savoir plus sur les manifestes des opérations par lot, consultez Spécification d'un manifeste.

   Rapport de fin de tâche

   Les opérations par lot S3 exécutent une seule tâche pour chaque objet spécifié dans le manifeste. Les rapports de fin de tâche fournissent un moyen facile de consulter les résultats de vos tâches dans un format consolidé sans nécessiter de configuration supplémentaire. Vous pouvez demander un rapport de fin pour toutes les tâches ou uniquement pour les tâches qui ont échoué. Pour en savoir plus sur les rapports de fin, consultez Rapports de fin de tâche.

   Autorisations

   L'une des causes les plus fréquentes des échecs de réplication est le manque d'autorisations dans le rôle fourni AWS Identity and Access Management (IAM). Pour plus d'informations sur la création de ce rôle, consultez Configuration des politiques IAM pour la réplication par lot.

7. Choisissez Create Batch Operations job (Créer une tâche d'opérations par lot).

Créer une tâche de réplication par lot pour les règles de réplication existantes

Vous pouvez configurer la réplication par lots S3 pour une configuration de réplication existante à l'aide AWS des SDK, AWS Command Line Interface (AWS CLI) ou de la console Amazon S3. Pour obtenir une présentation de la réplication par lot, consultez Réplication d'objets existants via la réplication par lot S3.

Comme condition préalable, vous devez créer un rôle Batch Operations AWS Identity and Access Management (IAM) pour accorder à Amazon S3 les autorisations nécessaires pour effectuer des actions en votre nom, voirConfiguration des politiques IAM pour la réplication par lot.

Lorsque la tâche de réplication par lot est terminée, vous recevez un rapport de fin d'opérations. Pour plus d'informations sur l'utilisation du rapport pour examiner la tâche, consultez Suivi de l'état de la tâche et des rapports de fin de tâche.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez Opérations par lot dans le panneau de navigation de la console Amazon S3.
3. Choisissez Create job (Créer une tâche).
4. Choisissez la Région dans laquelle vous souhaitez créer la tâche.
5. Sélectionnez le Manifest format (format du manifeste). Cet exemple montre comment créer un manifeste basé sur une configuration de réplication S3 existante.

> **Note**
>
> Le manifeste est la liste de tous les objets sur lesquels vous souhaitez exécuter l'action spécifiée. Pour en savoir plus sur les manifestes des opérations par lot, consultez Spécification d'un manifeste. Si un manifeste a été préparé, choisissez S3 inventory report (manifest.json) (Rapport d'inventaire S3 (manifest.json)) ou CSV. Si les objets de votre manifeste sont dans un compartiment versionné, vous devez spécifier les ID de version des objets. Pour plus d'informations sur la création d'un manifeste, consultez Spécification d'un manifeste.

6. Pour créer un manifeste basé sur votre configuration de la réplication, choisissez Create manifest using S3 Replication configuration (Créer un manifeste à l'aide de la configuration de réplication S3). Ajoutez ensuite le compartiment source à votre configuration de la réplication.
7. (Facultatif) Vous pouvez inclure des filtres supplémentaires tels que la date de création d'objets et le statut de la réplication. Pour obtenir des exemples de filtrage par statut de réplication, consultez Spécification d'un manifeste pour une tâche de réplication par lot.
8. Pour enregistrer un manifeste, sélectionnez Save Batch Operations manifest (Enregistrer le manifeste des opérations par lot).

   a. Si vous choisissez de générer et d'enregistrer un manifeste, vous devez choisir Bucket in this account (Compartiment dans ce compte) ou Bucket in another Compte AWS (Compartiment dans un autre). Spécifiez le nom du compartiment dans la zone de texte.

   > **Note**
   >
   > Le manifeste généré doit être stocké dans le même emplacement Région AWS que le compartiment source.

   b. Choisissez le Type de chiffrement.
9. (Facultatif) Fournissez une description.
10. Ajustez la Priority (priorité) de la tâche si nécessaire. Un nombre plus élevé est synonyme de priorité supérieure. Simple Storage Service (Amazon S3) tente d'exécuter des tâches à priorité supérieure avant les tâches à priorité inférieure. Pour plus d'informations sur la priorité des tâches, consultez Affectation d'une priorité de tâche.
11. (Facultatif) Générez un rapport de fin de tâche. Pour générer, sélectionnez Generate completion report (Générer un rapport de fin de tâche).

    Si vous choisissez de générer un rapport de fin de tâche, vous devez choisir de créer un rapport concernant Failed tasks only (uniquement les tâches ayant échoué) ou All tasks (toutes les tâches), et fournir un compartiment de destination pour le rapport.
12. Sélectionnez un rôle IAM valide.

> **Note**
>
> Pour plus d'informations sur la création d'un rôle IAM, consultez Configuration des politiques IAM pour la réplication par lot.

13. (Facultatif) Ajoutez des identifications de tâche à la tâche de réplication par lot.
14. Choisissez Next (Suivant).
15. Vérifiez votre configuration de la tâche et sélectionnez Create job (Créer une tâche).

Utilisation du AWS CLI avec un manifeste S3

L'exemple suivant crée une tâche de réplication par lots S3 à l'aide d'un manifeste généré par S3 pour le Compte AWS *111122223333*. Cet exemple va tenter de répliquer des objets existants et des objets qui n'ont pas pu être répliqués. Pour plus d'informations sur le filtrage par statut de réplication, consultez Spécification d'un manifeste pour une tâche de réplication par lot.

```
aws s3control create-job --account-id 111122223333 --operation
 '{"S3ReplicateObject":{}}' --report '{"Bucket":"arn:aws:s3:::***
 completion report bucket ****","Prefix":"batch-replication-report",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}'
 --manifest-generator '{"S3JobManifestGenerator": {"ExpectedBucketOwner":
 "111122223333", "SourceBucket": "arn:aws:s3:::*** replication source bucket
 ***", "EnableManifestOutput": false, "Filter": {"EligibleForReplication": true,
 "ObjectReplicationStatuses": ["NONE","FAILED"]}}}' --priority 1 --role-arn
 arn:aws:iam::111122223333:role/batch-Replication-IAM-policy --no-confirmation-required
 --region source-bucket-region
```

> **Note**
>
> La tâche doit être initiée à partir du même compartiment source de Région AWS réplication. Le rôle IAM `role/batch-Replication-IAM-policy` a déjà été créé. Consultez Configuration des politiques IAM pour la réplication par lot.

Une fois que vous avez lancé avec succès une tâche de réplication par lot, vous recevez l'ID de la tâche en tant que réponse. Vous pouvez surveiller cette tâche à l'aide de la commande suivante.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Utilisation du AWS CLI avec un manifeste fourni par l'utilisateur

L'exemple suivant permet de créer une tâche de réplication par lot S3 à l'aide d'un manifeste défini par l'utilisateur pour Compte AWS *111122223333*. Si les objets de votre manifeste sont dans

un compartiment versionné, vous devez spécifier les ID de version des objets. Seul l'objet dont l'ID de version est spécifié dans l'attaque de l'homme du milieu (HDM) sera répliqué. Pour plus d'informations sur la création d'un manifeste, consultez Spécification d'un manifeste.

```
aws s3control create-job --account-id 111122223333 --operation
 '{"S3ReplicateObject":{}}' --report '{"Bucket":"arn:aws:s3:::***
 completion report bucket ****","Prefix":"batch-replication-report",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}'
 --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key","VersionId"]},"Location":{"ObjectArn":"arn:aws:s3:::*** completion
 report bucket ****/manifest.csv","ETag":"Manifest Etag"}}' --priority 1 --role-arn
 arn:aws:iam::111122223333:role/batch-Replication-IAM-policy --no-confirmation-required
 --region source-bucket-region
```

> **Note**
>
> La tâche doit être initiée à partir du même compartiment source de Région AWS réplication. Le rôle IAM `role/batch-Replication-IAM-policy` a déjà été créé. Consultez Configuration des politiques IAM pour la réplication par lot.

Une fois que vous avez lancé avec succès une tâche de réplication par lot, vous recevez l'ID de la tâche en tant que réponse. Vous pouvez surveiller cette tâche à l'aide de la commande suivante.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-
bucket-region
```

Catégorisation de votre stockage à l'aide de balises

Utilisez le balisage des objets pour classer le stockage par catégorie. Chaque balise est une paire clés-valeurs.

Vous pouvez ajouter des balises à de nouveaux objets lorsque vous les chargez ou vous pouvez les ajouter à des objets existants.

- Vous pouvez associer jusqu'à 10 balises à un objet. Les balises associées à un objet doivent avoir des clés de balise uniques.
- Une clé d'étiquette peut comporter jusqu'à 128 caractères Unicode et les valeurs d'étiquette peuvent comporter jusqu'à 256 caractères Unicode. Les balises d'objet Amazon S3 sont

représentées en interne en UTF-16. Notez qu'en UTF-16, les caractères occupent une ou deux positions de caractère.

- La clé et les valeurs sont sensibles à la casse.
- Pour plus d'informations sur les restrictions relatives aux balises, consultez la section Restrictions relatives aux balises définies par l'utilisateur dans le guide de l'utilisateur AWS de Billing and Cost Management. Pour les restrictions de base relatives aux balises, consultez la section Restrictions relatives aux balises dans le guide de l'utilisateur Amazon EC2.

Exemples

Considérez les exemples de balisage suivants :

Example Informations PHI

Supposons qu'un objet contienne des données relatives aux infos santé protégées (PHI, Protected Health Information). Vous pouvez baliser l'objet à l'aide de la paire clé-valeur suivante.

```
PHI=True
```

ou

```
Classification=PHI
```

Example Fichiers de projet

Supposons que vous stockiez des fichiers de projet dans votre compartiment S3. Vous pouvez baliser ces objets avec une clé nommée `Project` et une valeur, comme illustré ci-après.

```
Project=Blue
```

Example Plusieurs balises

Vous pouvez ajouter plusieurs balises à un objet, comme illustré ci-après.

```
Project=x
Classification=confidential
```

Préfixes de nom de clé et balises

Les préfixes des noms des clés d'objet vous permettent également de classer le stockage. Cependant, le classement basé sur les préfixes est unidimensionnel. Examinez les noms de clés d'objet suivants :

```
photos/photo1.jpg
project/projectx/document.pdf
project/projecty/document2.pdf
```

Ces noms de clés ont les préfixes `photos/`, `project/projectx/` et `project/projecty/`. Ces préfixes permettent un classement par catégorie unidimensionnel. Autrement dit, tout élément sous un préfixe est une catégorie. Par exemple, le préfixe `project/projectx` identifie tous les documents liés au projet x.

Le balisage vous propose maintenant une autre dimension. Si vous souhaitez photo1 dans la catégorie de projet x, vous pouvez baliser l'objet en conséquence.

Autres avantages

Outre la classification des données, le balisage offre d'autres avantages, notamment :

- Les balises d'objets permettent un contrôle d'accès précis des autorisations. Par exemple, vous pouvez accorder à un utilisateur les autorisations de lire uniquement des objets avec des balises spécifiques.
- Les balises d'objets permettent une gestion du cycle de vie des objets précise dans laquelle vous pouvez spécifier un filtre basé sur des balises, en plus du préfixe de nom de clé, dans une règle de cycle de vie.
- Si vous utilisez Analyses Amazon S3, vous pouvez configurer des filtres pour regrouper des objets pour l'analyse par balises d'objets, par préfixe de nom de clé ou à la fois par préfixe et par balises.
- Vous pouvez également personnaliser CloudWatch les statistiques Amazon pour afficher les informations par le biais de filtres de balises spécifiques. Consultez les sections suivantes pour obtenir des détails.

> **Important**
>
> Il est acceptable d'utiliser des balises pour étiqueter des objets contenant des données confidentielles, par exemple, des informations personnelles identifiables ou des informations

santé protégées. Néanmoins, les balises à proprement dit ne doivent pas contenir d'informations confidentielles

Ajout d'ensembles de balises d'objet à plusieurs objets Amazon S3 avec une seule requête

Pour ajouter des ensembles de balises d'objets à plusieurs objets Amazon S3 avec une seule demande, vous pouvez utiliser la tâche d'opérations par lots S3. Vous fournissez à la fonctionnalité d'opérations par lot S3 une liste d'objets sur lesquels effectuer des opérations. La fonctionnalité des opérations par lot S3 appelle l'opération d'API respective pour effectuer l'opération spécifiée. Une tâche d'opérations par lot peut effectuer l'opération spécifiée sur des milliards d'objets contenant des exaoctets de données.

La fonctionnalité d'opérations par lot S3 suit la progression, envoie des notifications et stocke un rapport de fin détaillé sur toutes les actions, offrant ainsi une expérience sans serveur entièrement gérée et qui peut être vérifiée. Vous pouvez utiliser S3 Batch Operations via la console Amazon S3 AWS CLI, AWS les SDK ou l'API REST. Pour plus d'informations, consultez the section called "Principes de base des opérations par lot".

Pour en savoir plus sur les balises d'objet, consultez Gestion des balises d'objets.

Opérations API associées au balisage des objets

Amazon S3 prend en charge les opérations API suivantes qui sont spécifiques au balisage des objets :

Opérations d'API sur les objets

- PUT Object Tagging – Remplace les balises sur un objet. Vous spécifiez des balises dans le corps de la demande. Il existe deux scénarios distincts de gestion des balises d'objets à l'aide de cette API.
  - L'objet n'a aucune balise – A l'aide de cette API, vous pouvez ajouter un ensemble de balises à un objet (l'objet n'a pas de balises précédentes).
  - L'objet a un ensemble de balises existantes – Pour modifier l'ensemble de balises existant, vous devez tout d'abord le récupérer, le modifier côté client, puis utiliser cette API pour le remplacer.

> **Note**
>
> Si vous envoyez cette demande avec un ensemble de balises vide, Amazon S3 supprime l'ensemble de balises existant de l'objet. Si vous utilisez cette méthode, vous serez facturé pour une demande de Niveau 1 (PUT). Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
> Il est préférable d'utiliser la demande DELETE Object Tagging, car elle produit le même résultat sans entraîner de frais.

- GET Object Tagging – Retourne l'ensemble de balises associé à un objet. Amazon S3 retourne des balises d'objets dans le corps de la réponse.
- DELETE Object Tagging – Supprime l'ensemble de balises associé à un objet.

Autres opérations d'API prenant en charge le balisage

- PUT Object et Initiate Multipart Upload – Vous pouvez spécifier des balises lorsque vous créez des objets. Vous spécifiez les balises à l'aide de l'en-tête de demande `x-amz-tagging`.
- GET Object - Au lieu de retourner l'ensemble de balises, Amazon S3 retourne le nombre de balises d'objets dans l'en-tête `x-amz-tag-count` (uniquement si le demandeur dispose des autorisations de lecture des balises), car la taille de l'en-tête de réponse est limitée à 8 Ko. Si vous souhaitez afficher les balises, vous effectuez une autre demande pour l'opération d'API GET Object Tagging.
- POST Object – Vous pouvez spécifier les balises dans votre demande POST.

  Tant que les balises dans votre demande ne dépassent pas la limite de taille de l'en-tête de demande HTTP de 8 Ko, vous pouvez utiliser l'API `PUT Object` pour créer des objets avec des balises. Si les balises que vous spécifiez dépassent la limite de taille de l'en-tête, vous pouvez utiliser cette méthode POST dans le corps de laquelle vous incluez les balises.

  PUT Object - Copy – Vous pouvez spécifier `x-amz-tagging-directive` dans votre demande pour indiquer à Amazon S3 de copier (comportement par défaut) les balises ou de les remplacer par un nouvel ensemble de balises fourni dans la demande.

Remarques :

- Le balisage d'objet S3 est fortement cohérent. Pour de plus amples informations, veuillez consulter Modèle de cohérence des données Amazon S3.

Configurations supplémentaires

Cette section explique comment le balisage des objets est lié à d'autres configurations.

Balisage des objets et gestion du cycle de vie

Dans la configuration du cycle de vie du compartiment, vous pouvez spécifier un filtre pour sélectionner un sous-ensemble des objets auquel s'applique la règle. Vous pouvez spécifier un filtre basé sur les préfixes des noms de clés, les balises d'objets, ou les deux.

Supposons que vous stockiez des photos (aux formats brut et final) dans votre compartiment Amazon S3. Vous pouvez baliser ces objets comme illustré ci-après.

```
phototype=raw
or
phototype=finished
```

Vous pouvez envisager d'archiver les photos au format brut dans S3 Glacier quelque temps après leur création. Vous pouvez configurer une règle de cycle de vie avec un filtre qui identifie le sous-ensemble des objets avec le préfixe de nom de clé (`photos/`) qui ont une balise spécifique (`phototype=raw`).

Pour de plus amples informations, veuillez consulter Gestion du cycle de vie de votre stockage.

Balisage et réplication d'objets

Si vous avez configuré la réplication sur votre compartiment, Amazon S3 réplique les balises, à condition que vous ayez accordé à Amazon S3 l'autorisation de lire les balises. Pour plus d'informations, consultez Configuration de la réplication en direct.

Notifications d'événement d'identification d'objet

Vous pouvez configurer une notification d'événement Amazon S3 afin de recevoir une notification lorsqu'une étiquette d'objet est ajoutée ou supprimée d'un objet. Le type d'événement `s3:ObjectTagging:Put` vous avertit lorsqu'une étiquette est PUT sur un objet ou lorsqu'une étiquette existante est mise à jour. Le type d'événement `s3:ObjectTagging:Delete` vous avertit lorsqu'une étiquette est supprimée d'un objet. Pour en savoir plus, consultez Activation des notifications d'événement.

Pour plus d'informations sur le balisage d'objets, consultez les rubriques suivantes :

Rubriques

Stratégies de balisage et de contrôle d'accès

Vous pouvez également utiliser des stratégies d'autorisations (stratégies de compartiment et stratégies utilisateur) pour gérer les autorisations associées au balisage des objets. Pour les actions des stratégies, consultez les rubriques suivantes :

- Opérations sur les objets
- Opérations de compartiment

Les balises d'objets permettent un contrôle d'accès précis pour la gestion des autorisations. Vous pouvez accorder des autorisations conditionnelles basées sur les balises d'objets. Amazon S3 prend en charge les clés de condition suivantes que vous pouvez utiliser pour accorder des autorisations conditionnelles basées sur les balises d'objets.

- `s3:ExistingObjectTag/<tag-key>` – Utilisez cette clé de condition pour vérifier qu'une balise d'objet existante possède la clé de balise et la valeur spécifiques.

  > **Note**
  >
  > Lors de l'octroi des autorisations pour les opérations `PUT Object` et `DELETE Object`, cette clé de condition n'est pas prise en charge. Autrement dit, vous ne pouvez pas créer une stratégie pour accorder ou refuser à un utilisateur les autorisations de supprimer ou de remplacer un objet en fonction de ses balises existantes.

- `s3:RequestObjectTagKeys` – Utilisez cette clé de condition pour limiter les clés de balise que vous voulez autoriser sur les objets. Cela est utile lorsque vous ajoutez des balises à des objets à l'aide PutObjectTagging des PutObject requêtes d'objets and et POST.
- `s3:RequestObjectTag/<tag-key>` – Utilisez cette clé de condition pour limiter les clés de balise et valeurs que vous voulez autoriser sur les objets. Cela est utile lorsque vous ajoutez des balises à des objets à l'aide PutObjectTagging des PutObject requêtes and et POST Bucket.

Pour obtenir la liste complète des clés de condition spécifiques au service Amazon S3, veuillez consulter Exemples de politiques relatives aux compartiments utilisant des clés de condition. Les stratégies d'autorisations suivantes illustrent la façon dont le balisage des objets permet une gestion précise des autorisations d'accès.

Example 1 : autoriser un utilisateur à lire uniquement les objets qui ont une valeur de clé et une étiquette spécifiques

La politique d'autorisations suivante limite un utilisateur à la seule lecture des objets qui comportent la clé et la valeur d'étiquette `environment: production`. Cette politique utilise la clé de condition `s3:ExistingObjectTag` pour spécifier la clé et la valeur d'étiquette.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/JohnDoe"
      ]
    },
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:GetObjectVersion"],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "StringEquals":
        {"s3:ExistingObjectTag/environment": "production"}
    }
  }
  ]
}
```

Example 2 : restreindre les clés d'étiquette d'objet que les utilisateurs peuvent ajouter

La stratégie d'autorisations suivante accorde à un utilisateur les autorisations d'effectuer l'action `s3:PutObjectTagging`, ce qui lui permet d'ajouter des balises à un objet existant. La condition utilise la clé de condition `s3:RequestObjectTagKeys` pour spécifier les clés d'étiquette autorisées, telles que `Owner` ou `CreationDate`. Pour plus d'informations, consultez Création d'une condition avec plusieurs clés ou valeurs dans le Guide de l'utilisateur IAM.

Cette politique garantit que chaque clé d'étiquette spécifiée dans la demande est une clé d'étiquette autorisée. Le qualificateur `ForAnyValue` dans la condition garantit qu'au moins une des clés spécifiées doit être présente dans la demande.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Example 3 : exiger une clé et une valeur d'étiquette spécifiques pour permettre aux utilisateurs d'ajouter des étiquettes d'objet

L'exemple de politique suivant accorde à un utilisateur l'autorisation d'exécuter l'action `s3:PutObjectTagging`, qui permet à un utilisateur d'ajouter des étiquettes à un objet existant. La condition exige que l'utilisateur inclue une clé d'étiquette spécifique (telle que *`Project`*) avec la valeur définie sur *X*.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
         ]
```

```
        },
        "Effect": "Allow",
        "Action": [
          "s3:PutObjectTagging"
        ],
        "Resource": [
          "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        ],
        "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
          }
        }
      }
    ]
}
```

Gestion des balises d'objets

Cette section explique comment gérer les balises d'objets à l'aide AWS des SDK pour Java et .NET ou de la console Amazon S3.

Le balisage des objets vous permet de classer le stockage par catégorie. Chaque balise est une paire clé-valeur qui respecte les règles suivantes :

- Vous pouvez associer jusqu'à 10 balises à un objet. Les balises associées à un objet doivent avoir des clés de balise uniques.
- Une clé d'étiquette peut comporter jusqu'à 128 caractères Unicode et les valeurs d'étiquette peuvent comporter jusqu'à 256 caractères Unicode. Les balises d'objet Amazon S3 sont représentées en interne en UTF-16. Notez qu'en UTF-16, les caractères occupent une ou deux positions de caractère.
- La clé et les valeurs sont sensibles à la casse.

Pour en savoir plus sur les balises d’objet, consultez Catégorisation de votre stockage à l'aide de balises. Pour de plus amples informations sur les restrictions liées aux balises, consultez Restrictions encadrant les balises définies par l'utilisateur dans le Guide de l’utilisateur AWS Billing and Cost Management .

Utiliser la console S3.

Pour ajouter des balises à un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom du compartiment qui contient les objets auxquels vous souhaitez ajouter des balises.

   Vous pouvez également naviguer vers un dossier si vous le souhaitez.
3. Dans la liste Objets, cochez la case à côté des noms des objets auxquels vous souhaitez ajouter des balises.
4. Dans le menu Actions, choisissez Modifier les balises.
5. Vérifiez les objets répertoriés et choisissez Ajouter des balises.
6. Chaque balise d'objet est une paire clé-valeur. Saisissez une Key (Clé) et une Value (Valeur). Choisissez Add Tag (Ajouter une balise) pour ajouter une autre balise.

   Vous pouvez entrer jusqu'à 10 balises pour un objet.
7. Sélectionnez Enregistrer les modifications.

   Amazon S3 ajoute les balises aux objets spécifiés.

Pour de plus amples informations, veuillez consulter aussi Affichage des propriétés d'un objet dans la console Amazon S3 et Chargement d'objets dans ce guide.

Utilisation des AWS SDK

Java

L'exemple suivant montre comment utiliser le AWS SDK for Java pour définir des balises pour un nouvel objet et récupérer ou remplacer des balises pour un objet existant. Pour plus d'informations sur le balisage des objets, consultez Catégorisation de votre stockage à l'aide de balises. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Object key ***";
        String filePath = "*** File path ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Create an object, add two new tags, and upload the object to Amazon
 S3.
            PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
 new File(filePath));
            List<Tag> tags = new ArrayList<Tag>();
            tags.add(new Tag("Tag 1", "This is tag 1"));
            tags.add(new Tag("Tag 2", "This is tag 2"));
            putRequest.setTagging(new ObjectTagging(tags));
            PutObjectResult putResult = s3Client.putObject(putRequest);

            // Retrieve the object's tags.
            GetObjectTaggingRequest getTaggingRequest = new
 GetObjectTaggingRequest(bucketName, keyName);
            GetObjectTaggingResult getTagsResult =
 s3Client.getObjectTagging(getTaggingRequest);

            // Replace the object's tags with two new tags.
            List<Tag> newTags = new ArrayList<Tag>();
            newTags.add(new Tag("Tag 3", "This is tag 3"));
            newTags.add(new Tag("Tag 4", "This is tag 4"));
```

```
            s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName,
 keyName, new ObjectTagging(newTags)));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

L'exemple suivant montre comment utiliser le AWS SDK for .NET pour définir les balises d'un nouvel objet et récupérer ou remplacer les balises d'un objet existant. Pour plus d'informations sur le balisage des objets, consultez Catégorisation de votre stockage à l'aide de balises.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for the new object ***";
        private const string filePath = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
```

```
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1"},
                        new Tag { Key = "Keyx2", Value = "Value2" }
                    }
                };

                PutObjectResponse response = await
client.PutObjectAsync(putRequest);
                // 2. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
                for (int i = 0; i < objectTags.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
objectTags.Tagging[i].Key, objectTags.Tagging[i].Value);


                // 3. Replace the tagset.

                Tagging newTagSet = new Tagging();
                newTagSet.TagSet = new List<Tag>{
                    new Tag { Key = "Key3", Value = "Value3"},
```

```
                    new Tag { Key = "Key4", Value = "Value4" }
                };


                PutObjectTaggingRequest putObjTagsRequest = new
 PutObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                    Tagging = newTagSet
                };
                PutObjectTaggingResponse response2 = await
 client.PutObjectTaggingAsync(putObjTagsRequest);

                // 4. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest2 = new
 GetObjectTaggingRequest();
                getTagsRequest2.BucketName = bucketName;
                getTagsRequest2.Key = keyName;
                GetObjectTaggingResponse objectTags2 = await
 client.GetObjectTaggingAsync(getTagsRequest2);
                for (int i = 0; i < objectTags2.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
 objectTags2.Tagging[i].Key, objectTags2.Tagging[i].Value);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
 object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Encountered an error. Message:'{0}' when writing an object"
                    , e.Message);
            }
        }
    }
}
```

Utilisation des balises de répartition des coûts pour les compartiments S3

Pour effectuer le suivi des coûts de stockage ou d'autres éléments pour des projets ou groupes de projets, étiquetez vos compartiments Amazon S3 à l'aide de balises de répartition des coûts. Une balise de répartition des coûts correspond à une paire clé-valeur que vous associez à un compartiment S3. Une fois que ces balises sont activées, AWS s'en sert pour organiser les coûts de vos ressources dans votre rapport de répartition des coûts. Les balises de répartition des coûts ne peuvent être utilisées que pour nommer les compartiments. Pour en savoir plus sur les balises utilisées pour la dénomination d'objets, consultez Catégorisation de votre stockage à l'aide de balises.

Le rapport de répartition des coûts répertorie l'utilisation d'AWS correspondant à votre compte par catégorie de produits et l'utilisateur de compte lié. Il contient les mêmes postes que le rapport de facturation détaillée (consultez Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3), ainsi que des colonnes supplémentaires pour vos clés de balises.

AWS fournit deux types de balises de répartition des coûts, une balise générée par AWS et des balises définies par l'utilisateur. AWS définit, crée et applique l'étiquette `createdBy` générée par AWS après un événement Amazon S3 CreateBucket. Vous pouvez définir, créer et appliquer des balises définies par l'utilisateur à votre compartiment S3.

Vous devez activer ces deux types de balises séparément dans la console de gestion des coûts et de la facturation pour qu'elles apparaissent dans vos rapports de facturation. Pour de plus amples informations sur les balises générées par AWS, veuillez consulter Balises de répartition des coûts générées par AWS.

- Pour créer des balises dans la console, veuillez consulter Affichage des propriétés d'un compartiment S3.
- Pour créer des balises à l'aide de l'API Amazon S3, veuillez consulter PutBucketTagging dans la Référence d'API Amazon Simple Storage Service.
- Pour créer des étiquettes à l'aide de la AWS CLI, veuillez consulter put-bucket-tagging dans la Référence des commandes de la AWS CLI.
- Pour de plus amples informations sur l'activation de balises, veuillez consulter Utilisation des étiquettes de répartition des coûts dans le Guide de l'utilisateur AWS Billing.

Balises de répartition des coûts définies par l'utilisateur

Une balise de répartition des coûts définie par l'utilisateur comprend les composants suivants :

- Clé de balise . il s'agit du nom de la balise. Par exemple, dans la balise projet/Trinity, projet est la clé. La balise est sensible à la casse et doit contenir entre 1 et 128 caractères Unicode.

- Valeur de balise. Il s'agit d'une chaîne obligatoire. Par exemple, dans la balise projet/Trinity, Trinity est la valeur. La valeur de clé est sensible à la casse et doit contenir entre 0 et 256 caractères Unicode.

Pour connaître les caractères autorisés dans les étiquettes définies par l'utilisateur et les autres restrictions, veuillez consulter la section Restrictions encadrant les étiquettes définies par l'utilisateur du Guide de l'utilisateur AWS Billing. Pour de plus amples informations sur les étiquettes définies par l'utilisateur, veuillez consulter la section Étiquettes de répartition des coûts définies par l'utilisateur du Guide de l'utilisateur AWS Billing.

Balises de compartiment S3

Chaque compartiment S3 possède un ensemble de balises. Un ensemble de balises contient toutes les balises associées au compartiment. Un ensemble de balises peut contenir jusqu'à 50 balises ou n'en contenir aucune. Dans un ensemble de balises, les clés doivent être uniques, contrairement aux valeurs. Par exemple, vous pouvez utiliser une valeur identique dans des ensembles de balises nommés projet/Trinity et centre-de-coûts/Trinity.

Si, dans un compartiment, vous ajoutez une balise qui dispose de la même clé qu'une balise existante, la nouvelle valeur remplace l'ancienne valeur.

AWS n'applique aucune signification sémantique à vos balises. Nous interprétons les balises de façon stricte, en tant que chaîne de caractères.

Pour ajouter, répertorier, modifier ou supprimer des étiquettes, vous pouvez utiliser la console Amazon S3, l'AWS Command Line Interface (AWS CLI) ou l'API Amazon S3.

Plus d'informations

- Utilisation des étiquettes de répartition des coûts dans le Guide de l'utilisateur AWS Billing.
- Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3
- AWS Billing rapports pour Amazon S3

Rapports de facturation et d'utilisation pour Amazon S3

> **Important**
>
> Le 13 mai 2024, nous avons commencé à déployer une modification visant à éliminer les frais pour les demandes non autorisées qui ne sont pas initiées par le propriétaire du compartiment. Une fois le déploiement de cette modification terminé, les propriétaires de compartiments n'auront jamais à payer de frais de demande ou de bande passante pour les demandes renvoyant des erreurs `AccessDenied` (HTTP`403 Forbidden`) lorsque ces demandes sont initiées en dehors de leur AWS compte individuel ou de leur AWS organisation. Pour plus d'informations sur la liste complète des codes HTTP `3XX` et de `4XX` statut qui ne seront pas facturés, consultezFacturation des réponses aux erreurs d'Amazon S3. Cette modification de facturation ne nécessite aucune mise à jour de vos applications et s'applique à tous les compartiments S3. Lorsque le déploiement de cette modification sera complètement terminé Régions AWS, nous mettrons à jour notre documentation.

Lorsque vous utilisez Amazon S3, vous n'avez pas à payer de frais initiaux ni à vous engager à stocker du contenu. Comme les autres Services AWS, vous payez au fur et à mesure et vous ne payez que pour ce que vous utilisez.

AWS fournit les rapports suivants pour Amazon S3 :

- Rapports de facturation : plusieurs rapports fournissant des vues détaillées de toutes les activités liées au produit Services AWS que vous utilisez, y compris Amazon S3. AWS facture toujours au propriétaire du compartiment S3 les frais Amazon S3, sauf si le compartiment a été créé en tant que compartiment Requester Pays. Pour en savoir plus sur les compartiments de ce type, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation. Pour en savoir plus sur les rapports de facturation, consultez AWS Billing rapports pour Amazon S3.
- Rapport d'utilisation – Résumé des activités pour un service spécifique, avec les totaux horaires, quotidiens ou mensuels. Vous voulez choisir le type d'utilisation et les opérations à inclure. Vous pouvez également choisir la manière dont les données sont regroupées. Pour plus d’informations, consultez AWS rapport d'utilisation pour Amazon S3.

Les rubriques suivantes fournissent des informations sur les rapports d'utilisation et de facturation pour Amazon S3.

Rubriques

AWS Billing rapports pour Amazon S3

Votre facture mensuelle AWS sépare vos informations d'utilisation et vos coûts par Service AWS fonction. Plusieurs AWS Billing rapports sont disponibles : le rapport mensuel, le rapport de répartition des coûts et les rapports de facturation détaillés. Pour plus d'informations sur l'affichage de vos rapports de facturation, consultez Affichage d'une facture dans le Guide de l'utilisateur AWS Billing .

Pour suivre votre AWS utilisation et fournir une estimation des frais associés à votre compte, vous pouvez configurer AWS Cost and Usage Reports. Pour plus d'informations, voir Qu'est-ce que c'est AWSCost and Usage Reports ? dans le Guide d'exportation de AWS données.

Vous pouvez également télécharger un rapport d'utilisation, qui fournit plus d'informations sur votre utilisation du stockage Amazon S3 que les rapports de facturation. Pour plus d'informations, consultez AWS rapport d'utilisation pour Amazon S3.

Le tableau suivant répertorie les coûts liés à l'utilisation d'Amazon S3.

Coûts d'utilisation d'Amazon S3

Opération payante	Commentaires
Stockage	Des frais de stockage d'objets dans vos compartiments S3 vous sont facturés. Le tarif qui vous est facturé dépend de la taille de vos objets, de la durée pendant laquelle vous les avez stockés au cours du mois et de la classe de stockage. Amazon S3 propose les classes de stockage suivantes : S3 Standard, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA pour les accès peu fréquents), S3 One Zone-IA, S3 Glacier Instant Retrieval,

Opération payante	Commentaires
	S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou Reduced Redundancy Storage (RRS). Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). Sachez que si le contrôle de version S3 est activé, vous êtes facturé pour chaque version d'un objet conservée. Pour plus d'informations sur la gestion des versions, consultez Fonctionnement de la gestion des versions S3.
Surveillance et automatisation	Vous payez mensuellement des frais de surveillance et d'automatisation par objet stocké dans la classe de stockage S3 Intelligent-Tiering. Cela permet de surveiller les modèles d'accès et de déplacer les objets dans les niveaux d'accès au sein de la dite classe.
Requêtes	Vous payez pour les demandes, par exemple, `GET` effectuées contre vos compartiments et objets S3. Les demandes liées au cycle de vie sont incluses. Les tarifs des demandes dépendent du type de demande que vous faites. Pour plus d'informations sur la tarification des demandes, consultez Tarification Amazon S3.
Extractions	Des frais vous sont facturés pour l'extraction d'objets stockés dans les espaces de stockage S3 Standard – Accès peu fréquent, S3 Unizone – Accès peu fréquent, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.

Opération payante	Commentaires
Suppressions anticipées	Si vous supprimez un objet stocké dans un espace de stockage S3 standard – Accès peu fréquent, S3 unizone – Accès peu fréquent, S3 Glacier Instant Retrieval , S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive avant expiration du délai de stockage minimum pour lequel vous vous êtes engagé, des frais de suppression anticipée vous sont facturés pour l'objet en question.
Gestion du stockage	Vous payez pour les fonctionnalités de gestion du stockage (Amazon S3 Inventory, analyse et marquage d'objets) activées sur les compartiments de votre compte.
Bande passante	Vous payez pour l'ensemble de la bande passante entrante et sortante dans Amazon S3, à l'exception des éléments ci-dessous : • Données transférées depuis Internet • Données transférées vers une instance Amazon Elastic Compute Cloud (Amazon EC2), lorsque l'instance se trouve dans le Région AWS même compartiment que le compartiment S3 • Données transférées vers Amazon CloudFront (CloudFront) Vous payez également des frais pour toutes les données transférées à l'aide d'Amazon S3 Transfer Acceleration.

Pour obtenir des informations détaillées sur les frais d'utilisation d'Amazon S3 pour le stockage, le transfert de données et les services, consultez la tarification d'Amazon S3 et les FAQ Amazon S3.

Pour plus d'informations sur la compréhension des codes et des abréviations utilisés dans les rapports de facturation et d'utilisation d'Amazon S3, consultezComprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3.

Plus d'informations

- AWS rapport d'utilisation pour Amazon S3
- Utilisation des balises de répartition des coûts pour les compartiments S3
- AWS Billing et gestion des coûts
- Tarification Amazon S3

AWS rapport d'utilisation pour Amazon S3

Lorsque vous téléchargez un rapport d'utilisation, vous pouvez demander à cumuler les données d'utilisation sur une base horaire, quotidienne ou mensuelle. Le rapport d'utilisation d'Amazon S3 répertorie les opérations par type d'utilisation et Région AWS. Pour des rapports plus détaillés sur l'utilisation de l'espace de stockage Amazon S3, téléchargez les rapports d'utilisation AWS générés de façon dynamique. Vous voulez choisir le type d'utilisation, les opérations et la période à inclure. Vous pouvez également choisir la manière dont les données sont regroupées. Pour plus d'informations sur les rapports d'utilisation, voir Rapport AWS d'utilisation dans le Guide de l'utilisateur de AWS Data Exports.

Le rapport d'utilisation Amazon S3 inclut les informations suivantes :

- Service – Amazon S3
- Operation – Opération effectuée sur votre compartiment ou objet. Pour obtenir une explication détaillée des opérations Amazon S3, veuillez consulter Opérations de suivi dans vos rapports d'utilisation.
- UsageType— L'une des valeurs suivantes :
  - Code qui identifie le type de stockage
  - Code qui identifie le type de demande
  - Code qui identifie le type d'extraction
  - Code qui identifie le type de transfert de données

  - Code qui identifie les suppressions anticipées dans les espaces de stockage S3 Intelligent-Tiering, S3 standard – Accès peu fréquent, S3 One Zone-Infrequent Access (S3 One Zone-IA), S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
  - `StorageObjectCount` – Nombre d'objets stockés dans un compartiment donné

  Pour obtenir une explication détaillée des types d'utilisation d'Amazon S3, veuillez consulter Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3.
- Resource – Nom du compartiment associé à l'utilisation répertoriée.
- StartTime— Heure de début de la journée à laquelle s'applique l'utilisation, en temps universel coordonné (UTC).
- EndTime— Heure de fin de la journée à laquelle s'applique l'utilisation, en temps universel coordonné (UTC).
- UsageValue— L'une des valeurs de volume suivantes. L'unité de mesure standard des données est le gigaoctet (Go). Toutefois, selon le service et le rapport, des téraoctets (To) peuvent apparaître à la place.
  - Nombre de requêtes au cours de la période spécifiée
  - Volume de données transférées
  - Volume de données stockées dans une heure donnée
  - Volume de données associées à des restaurations à partir d'un espace de stockage S3 Standard – Accès peu fréquent, S3 Unizone – Accès peu fréquent, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive

> **Tip**
>
> Pour plus d’informations sur les demandes reçues par Amazon S3 pour vos objets, activez la journalisation des accès serveur de vos compartiments. Pour plus d’informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.

Vous pouvez télécharger un rapport d'utilisation au format XML ou CSV (séparé par des virgules). Voici un exemple de rapport d'utilisation au format CSV ouvert dans un tableur.

Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	15309
AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	19062
AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-created3	6/1/2017 0:00	7/1/2017 0:00	68
AmazonS3	PutObjectForRepl	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRepl	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRepl	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRepl	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Pour plus d'informations, consultez Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3.

Téléchargement du rapport AWS d'utilisation

Vous pouvez télécharger un rapport d'utilisation sous forme de fichier XML ou CSV.

Pour télécharger le rapport d'utilisation

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de titre, choisissez votre nom d'utilisateur ou votre identifiant de compte, puis choisissez Billing and Cost Management.
3. Dans le volet de navigation, sélectionnez Rapports sur les coûts et l'utilisation.
4. Sous Rapport AWS d'utilisation, choisissez Créer un rapport d'utilisation.
5. Sur la page Télécharger le rapport d'utilisation, sélectionnez les paramètres suivants :
   - Services — Choisissez Amazon Simple Storage Service.
   - Usage Types (Types d'utilisation) – Pour obtenir une explication détaillée des types d'utilisation Amazon S3, veuillez consulter Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3.
   - Operation (Opération) – Pour obtenir une explication détaillée des opérations Amazon S3, veuillez consulter Opérations de suivi dans vos rapports d'utilisation.
   - Period (Période) – Période que le rapport doit couvrir.
   - Report Granularity (Niveau de détail du rapport) – Indiquez si vous souhaitez que le rapport intègre les sous-totaux par heure, jour ou mois.
6. Choisissez Télécharger, choisissez le format de téléchargement (rapport XML ou rapport CSV), puis suivez les instructions pour ouvrir ou enregistrer le rapport.

Plus d'informations

- Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3
- AWS Billing rapports pour Amazon S3

Comprendre vos rapports AWS de facturation et d'utilisation pour Amazon S3

> **Important**
>
> Le 13 mai 2024, nous avons commencé à déployer une modification visant à éliminer les frais pour les demandes non autorisées qui ne sont pas initiées par le propriétaire du compartiment. Une fois le déploiement de cette modification terminé, les propriétaires de compartiments n'auront jamais à payer de frais de demande ou de bande passante pour les demandes renvoyant des erreurs `AccessDenied` (HTTP`403 Forbidden`) lorsque ces demandes sont initiées en dehors de leur AWS compte individuel ou de leur AWS organisation. Pour plus d'informations sur la liste complète des codes HTTP `3XX` et de `4XX` statut qui ne seront pas facturés, consultezFacturation des réponses aux erreurs d'Amazon S3. Cette modification de facturation ne nécessite aucune mise à jour de vos applications et s'applique à tous les compartiments S3. Lorsque le déploiement de cette modification sera complètement terminé Régions AWS, nous mettrons à jour notre documentation.

Les rapports d'utilisation et de facturation Amazon S3 utilisent des codes et abréviations. Pour les types d'utilisation figurant dans le tableau ci-dessous *`regionregion1`*, remplacez et par *`region2`* les abréviations de cette liste :

- APE1 : Asie-Pacifique (Hong Kong)
- APN1 : Asie-Pacifique (Tokyo)
- APN2 : Asie-Pacifique (Séoul)
- APN3 : Asie-Pacifique (Osaka)
- APS1 : Asie-Pacifique (Singapour)
- APS2 : Asie-Pacifique (Sydney)
- APS3 : Asie-Pacifique (Mumbai)
- APS4 : Asie-Pacifique (Djakarta)

- APS5 : Asie-Pacifique (Hyderabad)
- APS6 : Asie-Pacifique (Melbourne)
- CAN1 : Canada (Centre)
- CAN2 : Canada Ouest (Calgary)
- CNN1 : Chine (Beijing)
- CNW1 : Chine (Ningxia)
- AFS1 : Afrique (Le Cap)
- EUC2 : Europe (Zurich)
- EUN1 : UE (Stockholm)
- EUS2 : Europe (Espagne)
- EUC1 : UE (Francfort)
- EU : UE (Irlande)
- EUS1 : Europe (Milan)
- EUW2 : UE (Londres)
- EUW3 : UE (Paris)
- ILC1 : Israël (Tel Aviv)
- MEC1 : Moyen-Orient (EAU)
- MES1 : Moyen-Orient (Bahreïn)
- SAE1 : Amérique du Sud (São Paulo)
- UGW1 : AWS GovCloud (US-Ouest)
- UGE1 : AWS GovCloud (USA Est)
- USE1 (ou pas de préfixe) : USA Est (Virginie du Nord)
- USE2 : USA Est (Ohio)
- USW1 : USA Ouest (Californie du Nord)
- USW2 : USA Ouest (Oregon)

Pour les types d'utilisation des points d'accès multirégionaux S3 dans le tableau suivant, remplacez *`regiongroup1`* et par *`regiongroup2`* les abréviations de cette liste :

- AP : Asie-Pacifique
- AU : Australie

- EU : Europe
- IN : Inde
- NA : Amérique du Nord
- SA : Amérique du Sud

Les groupes de régions sont des regroupements géographiques de plusieurs Régions AWS régions. Pour plus d'informations, consultez Régions et zones de disponibilité. Pour obtenir des informations sur la tarification par Région AWS, veuillez consulter Tarification Amazon S3.

La première colonne du tableau suivant répertorie les types d'utilisation qui apparaissent dans les rapports d'utilisation et de facturation. L'unité de mesure standard des données est le gigaoctet (Go). Toutefois, selon le service et le rapport, des téraoctets (To) peuvent apparaître à la place.

Types d'utilisation

Type d'utilisation	Unités	Granularité	Description
region1-*region2*-AWS-In-A Bytes	Go	Par heure	La quantité de données accélérées transférées *region1* vers *region2*
region1-*region2*-AWS-In-A Bytes-T1	Go	Par heure	La quantité de données accélérées T1 transférées *region1* depuis*region2*, où T1 fait référence aux CloudFront demandes adressées à des points de présence (POPs) aux États-Unis, en Europe et au Japon
region1-*region2*-AWS-In-A Bytes-T2	Go	Par heure	La quantité de données accélérées T2 transférées *region1* depuis*region2*, où T2 fait référence aux CloudFront demandes

Type d'utilisation	Unités	Granularité	Description
			adressées POPs à tous les autres emplacements AWS périphériques
region1-*region2*-AWS-In-Bytes	Go	Par heure	La quantité de données transférée *region1* vers *region2*
region1-*region2*-AWS-Out-ABytes	Go	Par heure	La quantité de données accélérées transférées *region1* de *region2*
region1-*region2*-AWS-Out-ABytes-T1	Go	Par heure	La quantité de données accélérées T1 transférées de *region1* à*region2*, T1 faisant référence aux CloudFront demandes adressées aux POP aux États-Unis, en Europe et au Japon
region1-*region2*-AWS-Out-ABytes-T2	Go	Par heure	La quantité de données accélérées T2 transférées de *region1* à*region2*, où T2 fait référence aux CloudFront demandes adressées aux POP dans tous les autres emplacements AWS périphériques
region1-*region2*-AWS-Out-Bytes	Go	Par heure	La quantité de données transférée de *region1* à *region2*

Type d'utilisation	Unités	Granularité	Description
region-BatchOperations-Jobs	Nombre	Par heure	Nombre de tâches S3 Batch Operations exécutées.
region-BatchOperations-Objects	Nombre	Par heure	Nombre d'opérations d'objets effectuées par S3 Batch Operations
region-Bulk-Retrieval-Bytes	Go	Par heure	Volume de données extraites à l'aide de demandes S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive en bloc
region-BytesDeleted-GDA	Go	Mensuel	La quantité de données supprimées par une `DeleteObject` opération du stockage S3 Glacier Deep Archive
region-BytesDeleted-GIR	Go	Mensuel	La quantité de données supprimées par une `DeleteObject` opération du stockage S3 Glacier Instant Retrieval.
region-BytesDeleted-GLACIER	Go	Mensuel	La quantité de données supprimées par une `DeleteObject` opération depuis le stockage de S3 Glacier Flexible Retrieval

Type d'utilisation	Unités	Granularité	Description
region-BytesDeleted-INT	Go	Mensuel	La quantité de données supprimées par une DeleteObject opération du stockage S3 Intelligent-Tiering
region-BytesDeleted-RRS	Go	Mensuel	La quantité de données supprimées par une DeleteObject opération du stockage à redondance réduite (RRS)
region-BytesDeleted-SIA	Go	Mensuel	La quantité de données supprimées par une DeleteObject opération du stockage S3 Standard-IA
region-BytesDeleted-STANDARD	Go	Mensuel	La quantité de données supprimées par une DeleteObject opération depuis le stockage S3 Standard
region-BytesDeleted-ZIA	Go	Mensuel	La quantité de données supprimées par une DeleteObject opération du stockage S3 One Zone-IA
region-C3DataTransfer-In-Bytes	Go	Par heure	La quantité de données transférée vers Amazon S3 depuis Amazon EC2 au sein du même Région AWS

Type d'utilisation	Unités	Granularité	Description
region-C3DataTransfer-Out-Bytes	Go	Par heure	La quantité de données transférées à partir d'Amazon S3 vers Amazon EC2 au sein de la même Région AWS
region-CloudFront-In-Bytes	Go	Par heure	La quantité de données transférées vers et Région AWS depuis une CloudFront distribution
region-CloudFront-Out-Bytes	Go	Par heure	La quantité de données transférée d'une CloudFront distribution Région AWS à une autre
region-DataTransfer-In-Bytes	Go	Par heure	Volume de données transférées vers Amazon S3 à partir d'internet
region-DataTransfer-Out-Bytes	Go	Par heure	Volume de données transférées depuis Amazon S3 vers internet[1]
region-DataTransfer-Regional-Bytes	Go	Par heure	La quantité de données transférée depuis Amazon S3 vers les AWS ressources de ce dernier Région AWS

Type d'utilisation	Unités	Granularité	Description
region-EarlyDelete-ByteHrs	Go-heures	Par heure	Volume d'utilisation du stockage, calculé au prorata, pour les objets supprimés de l'espace de stockage S3 Glacier Flexible Retrieval , avant expiration du délai d'engagement minimum de 90 jours[2]
region-EarlyDelete-GDA	Go-heures	Par heure	Volume d'utilisation du stockage, calculé au prorata, pour les objets supprimés de l'espace de stockage S3 Glacier Deep Archive, avant expiration du délai d'engagement minimum de 180 jours [2]
region-EarlyDelete-GIR	Go-heures	Par heure	Volume d'utilisation du stockage, calculé au prorata, pour les objets supprimés de l'espace de stockage S3 Glacier Instant Retrieval , avant expiration du délai d'engagement minimum de 90 jours

Type d'utilisation	Unités	Granularité	Description
region-EarlyDelete-GIR-SmObjects	Go-heures	Par heure	Volume d'utilisation du stockage, calculé au prorata, pour les petits objets (de moins de 128 Ko) qui ont été supprimés de S3 Glacier Instant Retrieval avant expiration du délai d'engagement minimum de 90 jours
region-EarlyDelete-SIA	Go-heures	Par heure	Volume d'utilisation du stockage, au prorata, pour les objets supprimés dans le stockage S3 standard – Accès peu fréquent, avant expiration du délai d'engagement minimum de 30 jours[3]
region-EarlyDelete-SIA-SmObjects	Go-heures	Par heure	Volume d'utilisation du stockage, au prorata, pour les petits objets (de moins de 128 Ko) qui ont été supprimés du stockage S3 standard – Accès peu fréquent avant expiration du délai d'engagement minimum de 30 jours[3]

Type d'utilisation	Unités	Granularité	Description
region-EarlyDelete-ZIA	Go-heures	Par heure	Volume d'utilisation du stockage, au prorata, pour les objets supprimés dans le stockage S3 unizone – Accès peu fréquent, avant expiration du délai d'engagement minimum de 30 jours[3]
region-EarlyDelete-ZIA-SmObjects	Go-heures	Par heure	Volume d'utilisation du stockage, au prorata, pour les petits objets (de moins de 128 Ko) qui ont été supprimés du stockage S3 unizone – Accès peu fréquent avant expiration du délai d'engagement minimum de 30 jours[3]
region-Expedited-Retrieval-Bytes	Go	Par heure	Volume de données extraites à l'aide de demandes S3 Glacier Flexible Retrieval accélérées
region-Inventory-Objects Listed	Objets	Par heure	Nombre d'objets répertoriés pour un groupe d'objets (les objets sont regroupés par compartiment ou préfixe) avec une liste d'inventaire

Type d'utilisation	Unités	Granularité	Description
region-Monitoring-Automation-INT	Objets	Par heure	Nombre d'objets uniques surveillés et à niveau automatique de la classe de stockage S3 Intelligent-Tiering
region-MRAP-Out-Bytes	Go	Par heure	Quantité de données transférée via un point d'accès multirégional S3 hors des compartiments d'une région (tarification du routage des données MRAP).
region-MRAP-In-Bytes	Go	Par heure	Quantité de données transférée via un point d'accès multirégional S3 hors des compartiments d'une région (tarification du routage des données MRAP).
regiongroup1-regiongroup2--MRAP-Out-Bytes	Go	Par heure	La quantité de données transférée via un point d'accès multirégional S3 depuis un compartiment *regiongroup1* vers un client *regiongroup2* situé en dehors du AWS réseau.

Type d'utilisation	Unités	Granularité	Description
regiongroup1-regiongroup2--MRAP-In-Bytes	Go	Par heure	La quantité de données transférée via un point d'accès multirégional S3 vers un compartiment *regiongroup1* depuis un client *regiongroup2* situé en dehors du AWS réseau.
region-OverwriteBytes-Copy-GDA	Go	Mensuel	La quantité de données écrasées par une CopyObject opération depuis le stockage S3 Glacier Deep Archive
region-OverwriteBytes-Copy-GIR	Go	Mensuel	La quantité de données écrasées par une CopyObject opération depuis le stockage S3 Glacier Instant Retrieval.
region-OverwriteBytes-Copy-GLACIER	Go	Mensuel	La quantité de données écrasées par une CopyObject opération depuis le stockage S3 Glacier Flexible Retrieval
region-OverwriteBytes-Copy-INT	Go	Mensuel	La quantité de données écrasées par une CopyObject opération à partir du stockage S3 Intelligent-Tiering

Type d'utilisation	Unités	Granularité	Description
region-OverwriteBytes-Copy-RRS	Go	Mensuel	La quantité de données écrasées par une `CopyObject` opération à partir du stockage à redondance réduite (RRS)
region-OverwriteBytes-Copy-SIA	Go	Mensuel	La quantité de données écrasées par une `CopyObject` opération à partir du stockage S3 Standard-IA
region-OverwriteBytes-Copy-STANDARD	Go	Mensuel	La quantité de données écrasées par une `CopyObject` opération depuis le stockage standard S3
region-OverwriteBytes-Copy-ZIA	Go	Mensuel	La quantité de données écrasées par une `CopyObject` opération à partir du stockage S3 One Zone-IA
region-OverwriteBytes-Put-GDA	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération depuis le stockage S3 Glacier Deep Archive
region-OverwriteBytes-Put-GIR	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération depuis le stockage S3 Glacier Instant Retrieval.

Type d'utilisation	Unités	Granularité	Description
region-OverwriteBytes-Put-GLACIER	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération depuis le stockage S3 Glacier Flexible Retrieval
region-OverwriteBytes-Put-INT	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération à partir du stockage S3 Intelligent-Tiering
region-OverwriteBytes-Put-RRS	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération à partir du stockage à redondance réduite (RRS)
region-OverwriteBytes-Put-SIA	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération à partir du stockage S3 Standard-IA
region-OverwriteBytes-Put-STANDARD	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération depuis le stockage standard S3
region-OverwriteBytes-Put-ZIA	Go	Mensuel	La quantité de données écrasées par une `PutObject` opération à partir du stockage S3 One Zone-IA

Type d'utilisation	Unités	Granularité	Description
region1-*region2*-S3RTC-In-Bytes	Go	Mensuel	La quantité de données transférée pour le contrôle du temps de réplication S3 (S3 RTC) depuis *region2* et *region1* par les opérations GetObjectReplTime, PutObjectReplTime, InitiateMultipartUploadReplTime, UploadPartReplTime, CompleteMultipartUploadReplTime, et WriteACLReplTime

Type d'utilisation	Unités	Granularité	Description
region1-*region2*-S3RTC-Out-Bytes	Go	Mensuel	La quantité de données transférée pour le contrôle du temps de réplication S3 (S3 RTC) depuis *region1* et *region2* par les GetObjectReplTime opérationsPutObject ReplTime ,InitiateM ultipartU ploadRepl Time ,UploadPar tReplTime ,CompleteM ultipartU ploadReplTime , et WriteACLReplTime
region-Requests-GDA-Tier1	Nombre	Par heure	Le nombre dePUT,,COPY, POST CreateMul tipartUpl oad UploadPart , ou de CompleteM ultipartUpload requêtes sur les objets S3 Glacier Deep Archive [6]
region-Requests-GDA-Tier2	Nombre	Par heure	Le nombre d'objets S3 Glacier Deep Archive GET et les HEAD requêtes relatives à ces objets
region-Requests-GDA-Tier3	Nombre	Par heure	Nombre de demandes de restauration standard S3 Glacier Deep Archive

Type d'utilisation	Unités	Granularité	Description
region-Requests-GDA-Tier5	Nombre	Par heure	Nombre de demandes de restauration S3 Glacier Deep Archive en bloc
region-Requests-GIR-Tier1	Nombre	Par heure	Le nombre d'PUTobjets S3 Glacier Instant Retrieval ou de POST requêtes sur ces objets. COPY
region-Requests-GIR-Tier2	Nombre	Par heure	Le nombre de demandes non liées à S3 Glacier Instant Retrieval-Tier1 GET et toutes les autres requêtes sur des objets S3 Glacier Instant Retrieval.
region-Requests-GLACIER-Tier1	Nombre	Par heure	Le nombre dePUT,,, COPY POST CreateMultipartUpload UploadPart , ou de CompleteMultipartUpload requêtes sur les objets S3 Glacier Flexible Retrieval 6
region-Requests-GLACIER-Tier2	Nombre	Par heure	Le nombre de demandes non répertoriées sur les objets S3 Glacier Flexible Retrieval GET et toutes les autres demandes non répertoriées

Type d'utilisation	Unités	Granularité	Description
region-Requests-INT-Tier1	Nombre	Par heure	Le nombre de ou PUT de POST requêtes sur COPY les objets S3 Intelligent-Tiering
region-Requests-INT-Tier2	Nombre	Par heure	Le nombre de demandes non de niveau 1 GET et toutes les autres requêtes pour les objets S3 Intelligent-Tiering
region-Requests-SIA-Tier1	Nombre	Par heure	Le nombre d'objets PUT S3 Standard-IA ou de POST requêtes sur ces objets COPY
region-Requests-SIA-Tier2	Nombre	Par heure	Le nombre de demandes non liées à S3 Glacier Instant Retrieval-Tier1 GET et toutes les autres requêtes sur des objets S3 Standard-IA
region-Requests-Tier1	Nombre	Par heure	Le nombre de PUT ou de POST demandes pour S3 Standard, RRS et tags, ainsi que les LIST demandes pour tous les buckets et objets COPY
region-Requests-Tier2	Nombre	Par heure	Le nombre de demandes non liées au niveau 1 GET et toutes les autres

Type d'utilisation	Unités	Granularité	Description
region-Requests-Tier3	Nombre	Par heure	Nombre de demandes de cycle de vie envoyées à S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive et de demandes de restauration standard S3 Glacier Flexible Retrieval
region-Requests-Tier4	Nombre	Par heure	Nombre de transitio ns du cycle de vie vers le stockage S3 Glacier Instant Retrieval , S3 Intelligent-Tiering, S3 Standard – Accès peu fréquent ou S3 Unizone – Accès peu fréquent
region-Requests-Tier5	Nombre	Par heure	Nombre de demandes de restauration S3 Glacier Flexible Retrieval en bloc
region-Requests-Tier6	Nombre	Par heure	Nombre de demandes de restauration S3 Glacier Flexible Retrieval accélérées
region-Requests-Tier8	Nombre	Par heure	Le nombre de demandes de subventions d'accès S3
region-Requests-XZ-Tier1	Nombre	Par heure	Le nombre de COPY requêtes PUT ou de requêtes sur des objets S3 Express One Zone

Type d'utilisation	Unités	Granularité	Description
region-Requests-XZ-Tier2	Nombre	Par heure	Le nombre de demandes non liées à S3 Express One Zone-Tier1 GET et toutes les autres requêtes sur des objets S3 Express One Zone
region-Requests-ZIA-Tier1	Nombre	Par heure	Le nombre d'objets PUT S3 One POST Zone-IA ou de requêtes sur ces objets COPY
region-Requests-ZIA-Tier2	Nombre	Par heure	Le nombre de demandes non liées à S3 One Zone-IA-Tier1 GET et toutes les autres requêtes sur des objets S3 One Zone-IA
region-Retrieval-GIR	Go	Par heure	Volume de données extraites du stockage S3 Glacier Instant Retrieval
region-Retrieval-SIA	Go	Par heure	Volume de données extraites du stockage S3 standard – Accès peu fréquent
region-Retrieval-XZ	Go	Par heure	La partie des données qui dépasse 512 Ko dans une demande de récupération donnée (PUTouCOPY) avec le stockage S3 Express One Zone

Type d'utilisation	Unités	Granularité	Description
region-Retrieval-ZIA	Go	Par heure	Volume de données extraites du stockage S3 unizone – Accès peu fréquent
region-S3DSSE-In-Bytes	Go	Mensuel	La quantité de données cryptées à deux reprises par Amazon S3
region-S3DSSE-Out-Bytes	Go	Mensuel	La quantité de données à double chiffrement déchiffrées par Amazon S3
region-S3G-DataTransfer-In-Bytes	Go	Par heure	Volume de données transférées vers Amazon S3 pour restaurer des objets à partir d'un espace de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
region-S3G-DataTransfer-Out-Bytes	Go	Par heure	Volume de données transférées depuis Amazon S3 pour transmettre des objets vers un espace de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive

Type d'utilisation	Unités	Granularité	Description
region-Select-Returned-Bytes	Go	Par heure	Volume de données retournées avec des demandes Select dans un stockage S3 Standard
region-Select-Returned-GIR-Bytes	Go	Par heure	Volume de données retournées avec des demandes Select du stockage S3 Glacier Instant Retrieval
region-Select-Returned-INT-Bytes	Go	Par heure	Volume de données retournées avec des demandes Select dans un stockage S3 Intelligent-Tiering
region-Select-Returned-SIA-Bytes	Go	Par heure	Volume de données retournées avec des demandes Select dans un stockage S3 standard – Accès peu fréquent
region-Select-Returned-ZIA-Bytes	Go	Par heure	Volume de données retournées avec des demandes Select dans un stockage S3 unizone – Accès peu fréquent
region-Select-Scanned-Bytes	Go	Par heure	Volume de données analysées avec des demandes Select dans un stockage S3 Standard

Type d'utilisation	Unités	Granularité	Description
region-Select-Scanned-GIR-Bytes	Go	Par heure	Volume de données analysées avec des demandes Select du stockage S3 Glacier Instant Retrieval
region-Select-Scanned-INT-Bytes	Go	Par heure	Volume de données analysées avec les demandes Select d'un stockage S3 Intelligent-Tiering
region-Select-Scanned-SIA-Bytes	Go	Par heure	Volume de données analysées avec des demandes Select dans un stockage S3 standard – Accès peu fréquent
region-Select-Scanned-ZIA-Bytes	Go	Par heure	Volume de données analysées avec des demandes Select dans un stockage S3 unizone – Accès peu fréquent
region-Standard-Retrieval-Bytes	Go	Par heure	Volume de données extraites à l'aide de demandes S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive standard
region-StorageAnalytics-ObjCount	Objets	Par heure	Nombre d'objets uniques surveillés dans chaque configuration Analyse de classe de stockage.

Type d'utilisation	Unités	Granularité	Description
region-StorageLens-ObjCount	Objets	Chaque jour	Nombre d'objets uniques dans chaque tableau de bord S3 Storage Lens qui sont suivis par des métriques avancées et des recommandations S3 Storage Lens.
region-StorageLensFreeTier-ObjCount	Objets	Chaque jour	Nombre d'objets uniques dans chaque tableau de bord S3 Storage Lens qui sont suivis par des métriques d'utilisation S3 Storage Lens.
StorageObjectCount	Nombre	Chaque jour	Nombre d'objets stockés dans un compartiment donné
region-TagStorage-TagHrs	Balise-heures	Chaque jour	Nombre total de balises sur l'ensemble des objets du compartiment, par heure
region-TimedStorage-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage standard S3
region-TimedStorage-GDA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage S3 Glacier Deep Archive

Type d'utilisation	Unités	Granularité	Description
region-TimedStorage-GDA-Staging	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage intermédiaire de S3 Glacier Deep Archive
region-TimedStorage-GIR-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage S3 Glacier Instant Retrieval.
region-TimedStorage-GIR-SmObjects	Go par mois	Chaque jour	Nombre de Go par mois pendant lesquels de petits objets (inférieurs à 128 Ko) ont été stockés dans le stockage S3 Glacier Instant Retrieval.
region-TimedStorage-GlacierByteHrs	Go par mois	Chaque jour	Le nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage flexible de S3 Glacier Retrieval
region-TimedStorage-GlacierStaging	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage intermédiaire de S3 Glacier Flexible Retrieval

Type d'utilisation	Unités	Granularité	Description
region-TimedStorage-INT-FA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le niveau Frequent Access du stockage S3 Intelligent-Tiering 5
region-TimedStorage-INT-IA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le niveau d'accès peu fréquent du stockage S3 Intelligent-Tiering
region-TimedStorage-INT-AA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le niveau Archive Access du stockage S3 Intelligent-Tiering
region-TimedStorage-INT-AIA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le niveau Archive Instant Access du stockage S3 Intelligent-Tiering
region-TimedStorage-INT-DAA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le niveau Deep Archive Access du stockage S3 Intelligent-Tiering

Type d'utilisation	Unités	Granularité	Description
region-TimedStorage-RRS-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage à redondance réduite (RRS)
region-TimedStorage-SIA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage S3 Standard-IA
region-TimedStorage-SIA-SmObjects	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels de petits objets (inférieurs à 128 Ko) ont été stockés dans le stockage S3 Standard-IA 4
region-TimedStorage-XZ-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage S3 Express One Zone
region-TimedStorage-ZIA-ByteHrs	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels les données ont été stockées dans le stockage S3 One Zone-IA
region-TimedStorage-ZIA-SmObjects	Go par mois	Chaque jour	Nombre de Go/mois pendant lesquels de petits objets (inférieurs à 128 Ko) ont été stockés dans le stockage S3 One Zone-IA

Type d'utilisation	Unités	Granularité	Description
region-Upload-XZ	Go	Par heure	La quantité de données supérieure à 512 Ko dans une demande de téléchargement donnée (PUTouCOPY) avec S3 Express One Zone

Remarques

1. Si vous interrompez un transfert avant qu'il ne soit terminé, la quantité de données transférée peut dépasser la quantité de données que votre application reçoit. Cet écart peut se produire parce qu'une demande de fin de transfert ne peut pas être exécutée instantanément et qu'une certaine quantité de données peut être en transit, en attendant l'exécution de la demande de résiliation. Ces données en transit sont facturées comme des données transférées « sortantes ».
2. Lorsque des objets archivés dans la classe de stockage S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive sont supprimés, remplacés ou transférés vers une autre classe de stockage avant la fin de l'engagement de stockage minimum, qui est de 90 jours pour S3 Glacier Instant Retrieval et S3 Glacier Flexible Retrieval, ou de 180 jours pour S3 Glacier Deep Archive, des frais proportionnels par gigaoctet sont facturés pour les jours restants.
3. Pour les objets stockés dans S3 Standard-IA ou S3 One Zone-IA, lorsqu'ils sont supprimés, remplacés ou transférés vers une autre classe de stockage avant 30 jours, des frais proportionnels par gigaoctet sont facturés pour les jours restants.
4. Pour les petits objets (inférieurs à 128 Ko) stockés dans S3 Standard-IA ou S3 One Zone-IA, lorsqu'ils sont supprimés, remplacés ou transférés vers une autre classe de stockage avant 30 jours, des frais proportionnels par gigaoctet sont facturés pour les jours restants.
5. Il n'existe pas de taille d'objet facturable minimale pour les objets de la classe de stockage S3 Intelligent-Tiering. Les objets dont la taille est inférieure à 128 Ko ne sont pas surveillés ni admissibles à la hiérarchisation automatique. Les objets plus petits sont stockés dans S3 Intelligent-Tiering –Accès fréquent.
6. Lorsque vous initiez une `CreateMultipartUpload` ou une `UploadPartCopy` demande auprès des classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, les demandes sont facturées aux taux de demande standard S3 jusqu'à ce que vous ayez terminé le téléchargement partitionné. `UploadPart` Une fois le téléchargement terminé, la

`CompleteMultipartUpload` demande unique est facturée au `PUT` tarif du stockage S3 Glacier de destination. Les parties du téléchargement partitionné en cours pour un `PUT` vers la classe de stockage S3 Glacier Flexible Retrieval sont facturées en tant que stockage intermédiaire S3 Glacier Flexible Retrieval aux tarifs de stockage standard S3 jusqu'à ce que le téléchargement soit terminé. De même, les parties de téléchargement partitionné en cours pour a `PUT` vers la classe de stockage S3 Glacier Deep Archive sont facturées en tant que stockage intermédiaire S3 Glacier Deep Archive aux taux de stockage standard S3 jusqu'à ce que le téléchargement soit terminé.

7. S3 Express One Zone applique des frais fixes par demande pour des tailles de demande allant jusqu'à 512 Ko. Des frais supplémentaires par Go sont appliqués pour les `PUT` demandes et les `GET` demandes portant sur la portion de demande supérieure à 512 Ko.
8. Pour plus d'informations sur les fonctionnalités prises en charge pour la classe de stockage S3 Express One Zone, consultez Fonctionnalités Amazon S3 non prises en charge par S3 Express One Zone.
9. Les types d'utilisation dont les unités sont facturées en Go sont calculés en octets dans les rapports d'utilisation.
10. Un Go par mois est obtenu en prenant le nombre total d'heures en Go, en les agrégeant au cours d'un mois, puis en divisant par le nombre d'heures du mois en question. Pour en savoir plus, consultez les questions fréquemment posées : Comment serai-je débité et facturé pour mon utilisation d'Amazon S3 ?

> **Note**
>
> En général, les propriétaires de compartiments S3 sont facturés pour les demandes comportant des réponses HTTP `200 OK` réussies et des réponses d'erreur `4XX` du client HTTP. Les propriétaires de compartiments ne sont pas facturés pour les réponses aux erreurs `5XX` du serveur HTTP, telles que les `503 Slow Down` erreurs HTTP. Pour plus d'informations sur les codes d'erreur S3 sous HTTP `3XX` et les codes d'`4XX`état non facturés, consultezFacturation des réponses aux erreurs d'Amazon S3. Pour plus d'informations sur les frais de facturation si votre compartiment est configuré en tant que compartiment Requester Pays, consultezFonctionnement du Paiement par le demandeur.

Opérations de suivi dans vos rapports d'utilisation

Les opérations décrivent l'action entreprise sur votre AWS objet ou compartiment par le type d'utilisation spécifié. Les opérations sont indiquées par des codes explicites comme `PutObject`

ou `ListBucket`. Pour savoir quelles actions prises sur votre compartiment ont généré un type d'utilisation spécifique, utilisez ces codes. Lorsque vous générez un rapport d'utilisation, vous pouvez baser votre rapport sur toutes les opérations ou sur une opération en particulier comme `GetObject`.

Plus d'informations

- AWS rapport d'utilisation pour Amazon S3
- AWS Billing rapports pour Amazon S3
- Tarification Amazon S3
- FAQ sur Amazon S3

Facturation des réponses aux erreurs d'Amazon S3

> **Important**
>
> Le 13 mai 2024, nous avons commencé à déployer une modification visant à éliminer les frais pour les demandes non autorisées qui ne sont pas initiées par le propriétaire du compartiment. Une fois le déploiement de cette modification terminé, les propriétaires de compartiments n'auront jamais à payer de frais de demande ou de bande passante pour les demandes renvoyant des erreurs `AccessDenied` (HTTP`403 Forbidden`) lorsque ces demandes sont initiées en dehors de leur AWS compte individuel ou de leur AWS organisation. La page actuelle affiche une liste complète des codes HTTP `3XX` et de `4XX` statut qui ne seront pas facturés. Cette modification de facturation ne nécessite aucune mise à jour de vos applications et s'applique à tous les compartiments S3. Lorsque le déploiement de cette modification sera complètement terminé Régions AWS, nous mettrons à jour notre documentation.

En général, les propriétaires de compartiments S3 sont facturés pour les demandes comportant des réponses HTTP `200 OK` réussies et des réponses d'erreur `4XX` du client HTTP. Les propriétaires de compartiments ne sont pas facturés pour les réponses aux erreurs `5XX` du serveur HTTP, telles que les `503 Slow Down` erreurs HTTP. Pour plus d'informations sur les frais de facturation si votre compartiment est configuré en tant que compartiment Requester Pays, consultezFonctionnement du Paiement par le demandeur.

Le tableau suivant répertorie les codes d'erreur spécifiques sous HTTP `3XX` et les codes `4XX` d'état qui ne sont pas facturés. Pour les compartiments configurés avec l'hébergement de sites Web,

les frais de demande et autres frais applicables continueront de s'appliquer lorsque S3 renvoie un document d'erreur personnalisé ou pour les redirections personnalisées.

> **Note**
>
> Pour `AccessDenied` (HTTP`403 Forbidden`), S3 ne facture pas le propriétaire du compartiment lorsque la demande est initiée en dehors du AWS compte individuel du propriétaire du compartiment ou de l' AWS organisation du propriétaire du compartiment.

Code de statut HTTP	Code d'erreur	Description du code d'erreur	
301 – Déplacé de façon permanente	PermanentRedirect	Le bucket auquel vous tentez d'accéder doit être adressé à l'aide du point de terminaison spécifié. Envoyez toutes les futures demandes à ce point de terminaison.	
	PermanentRedirectControlError	L'opération d'API à laquelle vous tentez d'accéder doit être traitée à l'aide du point de terminaison spécifié. Envoyez toutes les futures demandes à ce point de terminaison.	
307 Redirection temporaire	TemporaryRedirect	Vous êtes redirigé vers le bucket pendant la mise à jour du serveur DNS (Domain Name System).	

Code de statut HTTP	Code d’erreur	Description du code d'erreur
400 Requête erronée	AuthorizationHeaderMalformed	L'en-tête d'autorisation que vous avez fourni n'est pas valide.
	AuthorizationQueryParametersError	Les paramètres de requête d'autorisation que vous avez fournis ne sont pas valides.
	ExpiredToken	Le jeton fourni a expiré.
	IllegalLocationConstraintException	Vous essayez d'accéder à un bucket depuis une région différente de celle dans laquelle le bucket existe. Pour éviter cette erreur, utilisez l'`--region`option. Par exemple : `aws s3 cp` *`awsexample.txt`* `s3://` *`example-s3-bucket`*`/ --region` *`ap-east-1`* .

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	InvalidArgument	Cette erreur peut se produire dans les conditions suivantes : • L'argument spécifié n'était pas valide. • Il manquait un en-tête obligatoire à la demande. • L'argument spécifié était incomplet ou n'était pas au bon format. • L'argument spécifié doit avoir une longueur supérieure ou égale à 3.	
	InvalidDigest	La valeur Content-MD5 ou checksum que vous avez spécifiée n'est pas valide.	
	InvalidEncryptionAlgorithmError	La demande de chiffrement que vous avez spécifiée n'est pas valide. La valeur valide est `AES256`.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	InvalidRequest	Cette erreur peut se produire dans les conditions suivantes : • La version de signature utilisée dans la demande est incorrecte. Utiliser `AWS4-HMAC-SHA256` (Signature Version 4). • Un point d'accès ne peut être créé que pour un bucket existant. • Le point d'accès n'est pas dans un état permettant de le supprimer. • Un point d'accès ne peut être répertorié que pour un compartiment existant. • Le jeton suivant n'est pas valide. • Au moins une action doit être spécifiée dans une règle de cycle de vie. •	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
		Au moins une règle de cycle de vie doit être spécifiée. • Le nombre de règles de cycle de vie ne doit pas dépasser la limite autorisée de 1 000 règles. • La plage du `MaxResults` paramètre n'est pas valide. • Les requêtes SOAP doivent être effectuées via une connexion HTTPS. • Amazon S3 Transfer Acceleration n'est pas pris en charge pour les buckets dont les noms ne sont pas conformes au DNS. • Amazon S3 Transfer Acceleration n'est pas pris en charge pour les buckets dont le nom contient des points (.). •	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
		Le point de terminaison Amazon S3 Transfer Acceleration ne prend en charge que les demandes de style virtuel. • Amazon S3 Transfer Acceleration n'est pas configuré sur ce compartiment. • Amazon S3 Transfer Acceleration est désactivé sur ce compartiment. • Amazon S3 Transfer Acceleration n'est pas pris en charge sur ce compartiment. Pour obtenir de l'aide, contactez AWS Support. • Amazon S3 Transfer Acceleration ne peut pas être activé sur ce compartiment. Pour obtenir de l'aide, contactez AWS Support.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
		• Valeurs contradictoires fournies dans les en-têtes HTTP et les paramètres de requête. • Valeurs contradictoires fournies dans les en-têtes HTTP et les champs de formulaire POST. • CopyObject demande effectuée sur des objets d'une taille supérieure à 5 Go.	
	Demande SOAP non valide	Le corps de la requête SOAP n'est pas valide.	
	InvalidStorageClass	La classe de stockage que vous avez spécifiée n'est pas valide.	
	InvalidTag	Votre demande contient une balise saisie qui n'est pas valide. Par exemple, votre demande peut contenir des clés dupliquées, des clés ou des valeurs trop longues ou des balises système.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	InvalidToken	Le jeton fourni est mal formé ou n'est pas valide.	
	URI non valide	L'URI spécifié n'a pas pu être analysé.	
	KeyTooLongError	Votre clé est trop longue.	
	Erreur ACL malformée	L'ACL que vous avez fournie n'était pas bien formée ou n'a pas été validée par rapport à notre schéma publié.	
	Demande de poste mal formée	Le corps de votre requête POST n'est pas un multipart/form-data correctement formé.	
	XML mal formé	Le code XML que vous avez fourni n'était pas bien formé ou n'a pas été validé par rapport à notre schéma publié.	
	MaxPostPreDataLengthExceededError	Les champs de votre requête POST précédant le fichier de téléchargement étaient trop volumineux.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	MetadataTooLarge	Vos en-têtes de métadonnées dépassent la taille de métadonnées maximale autorisée.	
	MissingRequestBody Error	Vous avez envoyé un document XML vide sous forme de demande.	
	MissingSecurityHeader	Il manque un en-tête obligatoire dans votre demande.	
	NoLoggingStatusForKey	Il n'existe pas de sous-ressource d'état de journalisation pour une clé.	
	RequestHeaderSecti onTooLarge	L'en-tête de demande et les paramètres de requête utilisés pour que la demande dépasse les tailles maximales autorisées	
	UnexpectedContent	Cette demande contient du contenu non pris en charge.	
	UserKeyMustBeSpecified	La requête POST du bucket doit contenir le nom de champ spécifié. S'il est spécifié, vérifiez l'ordre des champs.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	IncorrectEndpoint	Le compartiment spécifié existe dans une autre région. Dirigez les demandes vers le point de terminaison approprié.	
403 Interdit	RequestTimeTooSkewed	La différence entre l'heure de la demande et celle du serveur est trop importante.	
	SignatureDoesNotMatch	La signature de demande calculée par le serveur ne correspond pas à la signature que vous avez fournie. Vérifiez votre clé d'accès AWS secrète et votre méthode de signature. Pour plus d'informations, consultez Authentification REST et Authentification SOAP.	
	NotSignedUp	Votre compte n'est pas inscrit au service Amazon S3. Vous devez vous inscrire avant de pouvoir utiliser Amazon S3. Vous pouvez vous inscrire à l'adresse suivante : https://aws.amazon.com/s3	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	InvalidSecurity	Les informations de sécurité fournies ne sont pas valides.	
	InvalidPayer	Tout accès à cet objet a été désactivé . Pour obtenir de l'aide supplémentaire, consultez la section Contactez-nous.	
	InvalidAccessKeyId	L'identifiant de clé d' AWS accès que vous avez fourni n'existe pas dans nos dossiers.	
	AccountProblem	Il y a un problème avec votre Compte AWS appareil qui empêche l'opération de se terminer correctem ent. Pour obtenir de l'aide supplémentaire, consultez la section Contactez-nous.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
	UnauthorizedAccessError	Applicable dans les régions chinoises uniquement. Renvoyé lorsqu'une demande est envoyée à un bucket qui ne possède pas de licence ICP. Pour plus d'informations, voir ICP Recordal.	
404 – Non trouvé	NoSuchUpload	Le téléchargement partitionné spécifié n'existe pas. L'ID de téléchargement n'est peut-être pas valide ou le téléchargement en plusieurs parties a peut-être été abandonné ou terminé.	
	NoSuchWebsiteConfiguration	Le bucket spécifié n'a pas de configuration de site Web.	
Méthode 405 non autorisée	MethodNotAllowed	La méthode spécifiée n'est pas autorisée pour cette ressource.	

Code de statut HTTP	Code d’erreur	Description du code d'erreur	
409 Conflit	BucketAlreadyExists	Le nom de compartiment demandé n'est pas disponible. L'espace de noms du bucket est partagé par tous les utilisateurs du système. Spécifiez un autre nom et réessayez.	
	InvalidBucketState	La demande n'est pas valide pour l'état actuel du compartiment.	
	OperationAborted	Une opération conditionnelle conflictuelle est actuellement en cours sur cette ressource. Essaie encore.	
411 Longueur requise	MissingContentLength	Vous devez fournir l'en-tête HTTP Content-Length.	
4.1.2 Échec de la condition préalable	RequestIsNotMultiPartContent	Une requête POST de bucket doit être du type de boîtier multipart/form-data.	

Filtrer et récupérer des données à l'aide d'Amazon S3 Select

Avec Amazon S3 Select, vous pouvez utiliser des instructions de langage de requête structurée (SQL) pour filtrer le contenu d'un objet Amazon S3 afin de récupérer uniquement le sous-ensemble de données dont vous avez besoin. En utilisant Amazon S3 Select pour filtrer ces données, vous

pouvez réduire la quantité de données transférées par Amazon S3, ce qui limite le coût et la latence de récupération de ces données.

Amazon S3 Select ne vous permet d'interroger qu'un seul objet à la fois. Il fonctionne sur un objet stocké au format CSV, JSON ou Apache Parquet au format. Il fonctionne également avec un objet compressé avec GZIP ou BZIP2 (pour les objets CSV et JSON uniquement) et un objet chiffré côté serveur. Vous pouvez spécifier les résultats au format CSV ou JSON et vous pouvez décider de la manière dont les enregistrements sont délimités dans le résultat.

Vous transmettez des expression SQL à Amazon S3 dans la demande. Amazon S3 Select prend en charge un sous-ensemble du langage SQL. Pour de plus amples informations sur les éléments SQL pris en charge par Amazon S3 Select, veuillez consulter Référence SQL pour Amazon S3 Select.

Vous pouvez effectuer des requêtes SQL à l'aide de la console Amazon S3, du AWS Command Line Interface (AWS CLI), de l'opération de l'`SelectObjectContent`API REST ou AWS des SDK.

> **Note**
>
> La console Amazon S3 limite la quantité de données renvoyées à 40 Mo. Pour récupérer davantage de données, utilisez l'API AWS CLI ou.

Exigences et limites

Les exigences pour l'utilisation d'Amazon S3 Select sont les suivantes :

- Vous devez bénéficier d'une autorisation `s3:GetObject` sur le l'objet que vous interrogez.
- Si l'objet que vous interrogez est doté d'un chiffrement côté serveur avec des clés fournies par le client (SSE-C), vous devez utiliser `https` et fournir la clé de chiffrement dans la demande.

Les limites suivantes s'appliquent lors de l'utilisation d'Amazon S3 Select :

- S3 Select ne peut interroger qu'un seul objet par demande.
- La longueur maximale d'une expression SQL est de 256 Ko.
- La longueur maximale d'un enregistrement dans l'entrée ou le résultat est de 1 Mo.
- Amazon S3 Select peut seulement émettre des données imbriquées à l'aide du format de sortie JSON.

- Vous ne pouvez pas interroger un objet stocké dans les classes de stockage S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive ou RRS (Reduced Redundancy Storage). Vous ne pouvez pas non plus interroger un objet stocké dans le niveau S3 Intelligent-Tiering Archive Access ou le niveau S3 Intelligent-Tiering Deep Archive Access. Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Des restrictions supplémentaires s'appliquent lors de l'utilisation d'Amazon S3 Select avec un Parquet objet :

- Amazon S3 Select prend en charge uniquement la compression en colonnes avec GZIP ou Snappy. Amazon S3 Select ne prend pas en charge la compression de l'objet entier pour un Parquet objet.
- Amazon S3 Select ne prend pas en charge la sortie Parquet. Vous devez spécifier le format de sortie CSV ou JSON.
- La taille maximale des groupes de rangées non compressés est de 512 Mo.
- Vous devez utiliser les types de données spécifiés dans le schéma de l'objet.
- Une sélection opérée dans un champ répété renvoie uniquement la dernière valeur.

Création d'une demande

Lorsque vous créez une demande, vous fournissez des informations détaillées sur l'objet interrogé à l'aide d'un objet `InputSerialization`. Vous fournissez des informations détaillées sur la manière dont les résultats doivent être retournés à l'aide d'un objet `OutputSerialization`. Vous incluez également l'expression SQL qu'Amazon S3 Select utilise pour filtrer la demande.

Pour plus d'informations sur la création d'une demande Amazon S3 Select, consultez SelectObjectContent dans la Référence d'API Amazon Simple Storage Service. Vous pouvez également consulter l'un des exemples de code SDK dans les sections suivantes.

Demandes utilisant des plages d'analyse

Amazon S3 Select vous permet d'analyser un sous-ensemble d'un objet en spécifiant une plage d'octets à interroger. Vous pouvez ainsi paralléliser l'analyse de l'objet entier en divisant la tâche en demandes Amazon S3 Select distinctes pour une série de plages d'analyse qui ne se chevauchent pas.

Les plages d'analyse n'ont pas besoin d'être alignées sur des limites d'enregistrement. Une demande de plage d'analyse Amazon S3 Select s'exécute sur la plage d'octets que vous spécifiez. Un enregistrement qui commence dans la plage d'analyse spécifiée mais qui s'étend au-delà de celle-ci est traité par la requête. Par exemple, la section suivante montre un objet Amazon S3 contenant une série d'enregistrements au format CSV délimité par des lignes :

```
A,B
C,D
D,E
E,F
G,H
I,J
```

Supposons que vous utilisiez le paramètre `ScanRange` d'Amazon S3 Select avec Début à (l'octet) 1 et Fin à (l'octet) 4. La plage d'analyse commence donc à « `,` » et l'analyse est exécutée jusqu'à la fin de l'enregistrement, à partir de `C`. Votre demande de plage d'analyse renverra le résultat `C, D` car c'est la fin de l'enregistrement.

Amazon S3 Select scan range demande la prise en charge Parquet des objets CSV (sans les séparateurs entre guillemets) ou JSON (en `LINES` mode uniquement). Les objets CSV et JSON doivent être non compressés. Pour les objets JSON et CSV de type ligne, quand une plage d'analyse est spécifiée dans le cadre d'une demande Amazon S3 Select, tous les enregistrements qui commencent dans la plage d'analyse sont traités. Pour les objets Parquet, tous les groupes de lignes qui commencent dans la plage d'analyse sont traités.

Les demandes de plage de numérisation Amazon S3 Select peuvent être utilisées avec l' AWS CLI API Amazon S3 et AWS les SDK. Vous pouvez utiliser le paramètre `ScanRange` dans la demande Amazon S3 Select pour cette fonction. Pour plus d'informations, veuillez consulter SelectObjectContent dans la Référence d'API Amazon Simple Storage Service.

Erreurs

Amazon S3 Select renvoie un code d'erreur et un message d'erreur associé en cas de problème lors de la tentative d'exécution d'une requête. Pour obtenir la liste des codes d'erreur et des descriptions, veuillez consulter la section Liste des codes d'erreur SELECT Object Content de la page Réponses d'erreur dans la Référence de l'API Amazon Simple Storage Service.

Pour plus d'informations sur Amazon S3 Select, consultez les rubriques suivantes.

Rubriques

Exemples d'utilisation d'Amazon S3 Select sur un objet

Vous pouvez utiliser S3 Select pour sélectionner le contenu d'un objet à l'aide de la console Amazon S3, de l'API REST et des AWS kits de développement logiciel.

Pour plus d'informations sur les fonctions SQL prises en charge pour S3 Select, consultez Fonctions SQL.

Utilisation de la console S3

Pour sélectionner du contenu à partir d'un objet dans la console Amazon S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Choisissez le compartiment qui contient l'objet dont vous souhaitez sélectionner le contenu, puis choisissez le nom de l'objet.
4. Choisissez Actions d'objet, puis Requête avec S3 Select.
5. Configurez Paramètres d'entrée en fonction du format de vos données d'entrée.
6. Configurez Paramètres de sortie en fonction du format de sortie que vous souhaitez recevoir.
7. Pour extraire des enregistrements de l'objet choisi, sous Requête SQL, saisissez les commandes SELECT et SQL. Pour plus d'informations sur la procédure d'écriture de commandes SQL, consultez Référence SQL pour Amazon S3 Select.
8. Après avoir saisi des requêtes SQL, choisissez Exécuter la requête SQL. Ensuite, sous Résultats de la requête, vous pouvez voir les résultats de vos requêtes SQL.

Utilisation de l'API REST

Vous pouvez utiliser les AWS SDK pour sélectionner le contenu d'un objet. Toutefois, si l'application l'exige, vous pouvez envoyer directement des demandes REST. Pour plus d'informations sur le format de demande et de réponse, consultez SelectObjectContent.

Utilisation des AWS kits de développement logiciel

Vous pouvez utiliser Amazon S3 Select pour sélectionner une partie du contenu d'un objet à l'aide de `selectObjectContent` cette méthode. Si cette méthode aboutit, elle renvoie les résultats de l'expression SQL.

Java

Le code Java suivant retourne la valeur de la première colonne de chaque enregistrement stocké dans un objet contenant les données stockées au format CSV. Il demande également que les messages `Progress` et `Stats` soient retournés. Vous devez fournir un nom de compartiment valide, ainsi qu'un objet contenant les donnes au format CSV.

Pour obtenir des instructions sur la création et le test d'un échantillon de travail, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentEventVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in
 the form of an
 * InputStream of records and write it to a file.
```

```
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-
path}";
    private static final String QUERY = "select s._1 from S3Object s";

    public static void main(String[] args) throws Exception {
        final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

        SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
 CSV_OBJECT_KEY, QUERY);
        final AtomicBoolean isResultComplete = new AtomicBoolean(false);

        try (OutputStream fileOutputStream = new FileOutputStream(new File
 (S3_SELECT_RESULTS_PATH));
             SelectObjectContentResult result =
 s3Client.selectObjectContent(request)) {
            InputStream resultInputStream =
 result.getPayload().getRecordsInputStream(
                    new SelectObjectContentEventVisitor() {
                        @Override
                        public void visit(SelectObjectContentEvent.StatsEvent event)
                        {
                            System.out.println(
                                    "Received Stats, Bytes Scanned: " +
 event.getDetails().getBytesScanned()
                                            +  " Bytes Processed: " +
 event.getDetails().getBytesProcessed());
                        }

                        /*
                         * An End Event informs that the request has finished
 successfully.
                         */
                        @Override
                        public void visit(SelectObjectContentEvent.EndEvent event)
                        {
                            isResultComplete.set(true);
                            System.out.println("Received End Event. Result is
 complete.");
```

```
                        }
                    }
            );

            copy(resultInputStream, fileOutputStream);
        }

        /*
         * The End Event indicates all matching records have been transmitted.
         * If the End Event is not received, the results may be incomplete.
         */
        if (!isResultComplete.get()) {
            throw new Exception("S3 Select request was incomplete as End Event was
 not received.");
        }
    }

    private static SelectObjectContentRequest generateBaseCSVRequest(String bucket,
 String key, String query) {
        SelectObjectContentRequest request = new SelectObjectContentRequest();
        request.setBucketName(bucket);
        request.setKey(key);
        request.setExpression(query);
        request.setExpressionType(ExpressionType.SQL);

        InputSerialization inputSerialization = new InputSerialization();
        inputSerialization.setCsv(new CSVInput());
        inputSerialization.setCompressionType(CompressionType.NONE);
        request.setInputSerialization(inputSerialization);

        OutputSerialization outputSerialization = new OutputSerialization();
        outputSerialization.setCsv(new CSVOutput());
        request.setOutputSerialization(outputSerialization);

        return request;
    }
}
```

JavaScript

Pour un JavaScript exemple d'utilisation de l'opération AWS SDK for JavaScript avec l'`SelectObjectContent`API S3 pour sélectionner des enregistrements à partir de fichiers JSON

et CSV stockés dans Amazon S3, consultez le billet de blog Présentation de la prise en charge d'Amazon S3 Select dans le AWS SDK for JavaScript.

Python

Pour un exemple Python sur l'utilisation de requêtes SQL pour effectuer des recherches dans des données chargées sur Amazon S3 en tant que fichier CSV (valeur séparée par des virgules) à l'aide de S3 Select, consultez le billet de blog Interrogation de données sans serveur ou base de données à l'aide d'Amazon S3 Select.

Référence SQL pour Amazon S3 Select

Cette référence contient une description des éléments du langage de recherche structurée (SQL) pris en charge par Amazon S3 Select.

Rubriques

SELECT commande

Amazon S3 Select prend en charge uniquement la commande SQL `SELECT`. Les clauses standard ANSI suivantes sont prises en charge pour `SELECT`:

- `SELECT` liste
- `FROM`Clause
- `WHERE`Clause
- `LIMIT`Clause

> **Note**
>
> Les requêtes Amazon S3 Select ne prennent actuellement pas en charge les sous-requêtes ni les jointures.

SELECT liste

La liste SELECT nomme les colonnes, fonctions et expressions que la requête doit renvoyer. La liste représente le résultat de la requête.

```
SELECT *
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

Le premier format SELECT avec * (astérisque) renvoie chaque ligne ayant transmis la clause WHERE, telle qu'elle. Le second format SELECT crée une ligne avec la ***projection1*** et la ***projection2*** des expressions scalaires de sortie définies par l'utilisateur de chaque colonne.

FROMClause

Amazon S3 Select prend en charge les formes suivantes de la clause FROM :

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

Dans chaque forme de la clause FROM, `table_name` est le `S3Object` qui est interrogé. Les utilisateurs provenant de bases de données relationnelles traditionnelles peuvent penser qu'il s'agit d'un schéma de base de données contenant plusieurs vues sur une table.

Dans le langage SQL standard, la clause FROM crée des lignes qui sont filtrées dans la clause WHERE et projetées dans la liste SELECT.

Pour les objets JSON stockés dans Amazon S3 Select, vous pouvez également utiliser les formes suivantes de la clause FROM :

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

En utilisant cette forme de la clause FROM, vous pouvez effectuer une sélection dans les tableaux ou les objets contenus dans un objet JSON. Vous pouvez spécifier `path` en utilisant l'un des formats suivants :

- Par nom (dans un objet) : `.name` ou `['name']`
- Par index (dans un tableau)) : `[index]`
- Par caractère générique (dans un objet) : `.*`
- Par caractère générique (dans un tableau) : `[*]`

> **Note**
>
> - Cette forme de la clause FROM fonctionne uniquement avec les objets JSON.
> - Les caractères génériques émettent toujours au moins un enregistrement. Si aucun enregistrement ne correspond, Amazon S3 Select émet la valeur MISSING. Pendant la sérialisation de la sortie (après que la requête a fini de s'exécuter), Amazon S3 Select remplace les valeurs MISSING par des enregistrements vides.
> - Les fonctions d'agrégation (AVG, COUNT, MAX, MIN et SUM) ignorent les valeurs MISSING.
> - Si, lorsque vous utilisez un caractère générique, vous n'indiquez pas d'alias, vous pouvez faire référence à la ligne en utilisant le dernier élément du chemin. Par exemple, vous pouvez sélectionner tous les prix d'une liste de livres en utilisant la requête `SELECT price FROM S3Object[*].books[*].price`. Si le chemin se termine par un caractère générique plutôt qu'un nom, vous pouvez utiliser la valeur `_1` pour faire référence à la ligne. Par exemple, à la place de `SELECT price FROM S3Object[*].books[*].price`, vous pouvez utiliser la requête `SELECT _1.price FROM S3Object[*].books[*]`.
> - Amazon S3 Select traite toujours un document JSON comme un tableau de valeurs de niveau racine. Par conséquent, même si l'objet JSON que vous interrogez ne contient qu'un seul élément racine, la clause FROM doit commencer par `S3Object[*]`. Toutefois, pour des raisons de compatibilité, Amazon S3 Select vous autorise à omettre le caractère générique si vous n'incluez pas de chemin. Ainsi, la clause complète `FROM S3Object` équivaut à `FROM S3Object[*] as S3Object`. Si vous incluez un chemin, vous devez également utiliser le caractère générique. Autrement dit, `FROM S3Object` et `FROM S3Object[*].path` sont deux clauses valides, mais pas `FROM S3Object.path`.

Example

Exemples :

Exemple 1

Cet exemple affiche les résultats quand le jeu de données et la requête suivants sont utilisés :

```
{ "Rules": [ {"id": "1"}, {"expr": "y > x"}, {"id": "2", "expr": "z = DEBUG"} ]}
{ "created": "June 27", "modified": "July 6" }
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"1"}
{}
{"id":"2"}
{}
```

Amazon S3 Select génère chaque résultat pour les raisons suivantes :

- `{"id":"id-1"}` : `S3Object[0].Rules[0].id` a produit une correspondance.
- `{}` : `S3Object[0].Rules[1].id` n'a pas produit de correspondance. Amazon S3 Select a donc émis `MISSING`, qui a ensuite été changé en enregistrement vide pendant la sérialisation de la sortie avant d'être renvoyé.
- `{"id":"id-2"}` : `S3Object[0].Rules[2].id` a produit une correspondance.
- `{}` : `S3Object[1]` n'a pas produit de correspondance sur `Rules`. Amazon S3 Select a donc émis `MISSING`, qui a ensuite été changé en enregistrement vide pendant la sérialisation de la sortie avant d'être renvoyé.

Si vous ne voulez pas qu'Amazon S3 Select renvoie des enregistrements vides en l'absence de correspondance, vous pouvez faire un test avec la valeur `MISSING`. La requête suivante renvoie les mêmes résultats que la requête précédente, à la différence près que les valeurs vides sont omises :

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Exemple 2

Cet exemple affiche les résultats quand le jeu de données et les requêtes suivants sont utilisés :

```
{ "created": "936864000", "dir_name": "important_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": ".aws" }, { "name": "downloads" } ], "owner": "Amazon
 S3" }
{ "created": "936864000", "dir_name": "other_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3Object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":"."},{"name":".."},{"name":".aws"},
{"name":"downloads"}]}
{"dir_name":"other_docs","files":[{"name":"."},{"name":".."},{"name":"my stuff"},
{"name":"backup"}]}
```

```
SELECT _1.dir_name, _1.owner FROM S3Object[*]
```

```
{"dir_name":"important_docs","owner":"Amazon S3"}
{"dir_name":"other_docs","owner":"User"}
```

WHEREClause

La clause `WHERE` suit la syntaxe suivante :

```
WHERE condition
```

La clause `WHERE` filtre les lignes en fonction de la *`condition`*. Une condition est une expression générant un résultat booléen. Seules les lignes pour lesquelles la condition a la valeur `TRUE` sont renvoyées dans les résultats.

LIMITClause

La clause `LIMIT` suit la syntaxe suivante :

```
LIMIT number
```

La clause `LIMIT` limite le nombre d'enregistrements que la requête devra renvoyer en fonction du *`number`*.

Accès aux attributs

Les clauses `SELECT` et `WHERE` peuvent faire référence à des enregistrements à l'aide d'une des méthodes présentées dans les sections suivantes, la méthode utilisée variant selon le format du fichier interrogé (CSV ou JSON).

CSV

- Numéros de colonne : vous pouvez faire référence à la Nième colonne d'une ligne avec un nom de colonne *_N*, où *N* correspond à la position de la colonne. Le compte de la position commence à 1. Par exemple, la première colonne est appelée _1 et la deuxième colonne est appelée _2.

  Vous pouvez faire référence à une colonne sous la forme *_N* ou *alias*.*_N*. Par exemple, _2 et `myAlias._2` sont deux manières valides de faire référence à une colonne de la liste `SELECT` et la clause `WHERE`.
- En-têtes de colonne – Concernant les objets au format CSV ayant une ligne d'en-tête, les en-têtes sont disponibles pour la liste `SELECT` et la clause `WHERE`. Comme dans le SQL traditionnel, dans les expressions de clause `SELECT` et `WHERE`, vous pouvez faire référence aux colonnes via *alias*.*column_name* ou *column_name*.

JSON

- Document – Vous pouvez accéder aux champs du document JSON sous la forme *alias*.*name*. Vous pouvez aussi accéder au champs imbriqués ; par exemple, *alias*.*name1*.*name2*.*name3*.
- Liste : vous pouvez accéder aux éléments d'une liste JSON en utilisant des index de base zéro avec l'opérateur `[]`. Par exemple, vous pouvez accéder au second élément d'une liste sous la forme *alias*`[1]`. Vous pouvez combiner l'accès aux éléments de la liste avec des champs, par exemple, *alias*.*name1*.*name2*`[1]`.*name3*.
- Exemples : Considérez cet objet JSON comme un exemple d'ensemble de données :

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
    [
     {"project_name":"project1", "completed":false},
     {"project_name":"project2", "completed":true}
    ]
}
```

Exemple 1

La requête suivante renvoie ces résultats :

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Exemple 2

La requête suivante renvoie ces résultats :

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name":"project1"}
```

Sensibilité à la casse des noms d'en-tête et d'attribut

Avec Amazon S3 Select, vous pouvez utiliser des guillemets doubles pour indiquer que les en-têtes de colonne (pour les objets CSV) et les attributs (pour les objets JSON) sont sensibles à la casse. Sans guillemets doubles, les en-têtes et les attributs d'objet sont sensibles à la casse. Une erreur est envoyée en cas d'ambiguïté.

Les exemples suivants sont 1) des objets Amazon S3 au format CSV avec les en-têtes de colonne spécifiés et avec `FileHeaderInfo` défini sur `"Use"` pour la demande de requête ou 2) des objets Amazon S3 au format JSON avec les attributs spécifiés.

Exemple n°1 : l'objet interrogé possède l'en-tête ou l'attribut `NAME`.

- L'expression suivante renvoie avec succès les valeurs de l'objet. Comme il n'y a pas de guillemets, la requête est sensible à la casse.

```
SELECT s.name from S3Object s
```

- L'expression suivante renvoie une erreur 400 `MissingHeaderName`. Comme il y a des guillemets, la requête est sensible à la casse.

```
SELECT s."name" from S3Object s
```

Exemple n°2 : l'objet Amazon S3 interrogé possède un en-tête ou un attribut avec `NAME` et un autre en-tête ou attribut avec `name`.

- L'expression suivante renvoie une erreur 400 `AmbiguousFieldName`. Comme il n'y a pas de guillemets doubles, la requête est sensible à la casse, mais il y a deux correspondances : l'erreur est renvoyée.

  ```
  SELECT s.name from S3Object s
  ```

- L'expression suivante renvoie avec succès les valeurs de l'objet. Comme il y a des guillemets, la requête est sensible à la casse sans qu'il y ait d'ambiguïté.

  ```
  SELECT s."NAME" from S3Object s
  ```

Utilisation de mots-clés réservés comme termes définis par l'utilisateur

Amazon S3 Select dispose d'un ensemble de mots-clés réservés nécessaires pour exécuter les expressions SQL utilisées pour interroger le contenu d'un objet. Les mots-clés réservés peuvent être des noms de fonctions, des types de données, des opérateurs, etc. Dans certains cas, les termes définis par l'utilisateur comme les en-têtes de colonnes (pour les fichiers CSV) ou les attributs (pour les objets JSON) peuvent entrer en conflit avec un mot-clé réservé. Lorsque cela se produit, vous devez utiliser des guillemets doubles pour indiquer que vous utilisez intentionnellement un terme défini par l'utilisateur qui est en conflit avec un mot-clé réservé. Si vous ne le faites pas, un erreur d'analyse 400 est émise.

Pour obtenir la liste complète des mots-clés réservés, consultez Mots-clés réservés.

L'exemple suivant est 1) un objet Amazon S3 au format CSV avec les en-têtes de colonne spécifiés et avec `FileHeaderInfo` défini sur `"Use"` pour la demande de requête ou 2) un objet Amazon S3 au format JSON avec les attributs spécifiés.

Exemple : l'objet interrogé possède un en-tête ou un attribut `CAST`, qui est un mot-clé réservé.

- L'expression suivante renvoie avec succès les valeurs de l'objet. Comme les guillemets sont utilisés dans la requête, S3 Select utilise l'en-tête ou l'attribut défini par l'utilisateur.

```
SELECT s."CAST" from S3Object s
```

- L'expression suivante renvoie une erreur d'analyse 400. Comme aucun guillemet n'est utilisé dans la requête, CAST entre en conflit avec un mot-clé réservé.

```
SELECT s.CAST from S3Object s
```

Expressions scalaires

Au sein de la clause WHERE et de la liste SELECT, vous pouvez disposez d'expressions scalaires SQL, qui sont des expressions qui renvoient des valeurs scalaires. Elles sont au format suivant :

- ***literal***

  Littéral SQL.

- ***column_reference***

  Référence à une colonne au format *column_name* ou *alias*.*column_name*.

- ***unary_op expression***

  Dans ce cas, ***unary_op*** est un opérateur unaire SQL.

- ***expression binary_op expression***

  Dans ce cas, ***binary_op*** est un opérateur binaire SQL.

- ***func_name***

  Dans ce cas, ***func_name*** est le nom de la fonction scalaire à appeler.

- ***expression*** [NOT] BETWEEN ***expression*** AND ***expression***
- ***expression*** LIKE ***expression*** [ESCAPE ***expression***]

Types de données

Amazon S3 Select prend en charge plusieurs types de données primitifs.

Conversions du type de données

La règle générale consiste à suivre la fonction `CAST` si elle est définie. Si `CAST` n'est pas défini, toutes les données en entrée sont considérées comme une chaîne. Dans ce cas, vous devez convertir vos données d'entrée dans les types de données concernés au besoin.

Pour plus d'informations sur la fonction `CAST`, consultez CAST.

Types de données pris en charge

Amazon S3 Select prend en charge l'ensemble de types de données primitifs suivant.

Name (Nom)	Description	Exemples
`bool`	Une valeur booléenne, `TRUE` ou `FALSE`.	`FALSE`
`int`, `integer`	Un entier signé sur 8 octets dans la plage -9 223 372 036 854 775 808 à 9 223 372 036 854 775 807.	`100000`
`string`	Une chaîne de longueur variable encodée en UTF8. La limite par défaut est de un caractère. La limite de caractères maximale est de 2 147 483 647.	`'xyz'`
`float`	Un nombre à virgule flottante de 8 octets.	`CAST(0.456 AS FLOAT)`
`decimal`, `numeric`	Un nombre de base 10, avec une précision maximale de 38 (c'est-à-dire, le nombre maximal de chiffres significatifs) et avec une échelle dans plage de -2^{31} à $2^{31}-1$ (c'est-à-dire, l'exposant de base 10). **Note** Amazon S3 Select ignore l'échelle et la précision lorsque vous fournissez les deux en même temps.	`123.456`
`timestamp`		`CAST('2007-04-05T1`

Name (Nom)	Description	Exemples
	Les horodatages représentent un moment spécifique dans le temps, incluent toujours un décalage local, et sont capables de précision arbitraire. Dans le format de texte, les horodatages suivent la remarque de W3C sur les formats de date et d'heure, mais ils doivent se terminer par le `T` littéral si la précision des horodatages n'est pas d'une journée complète. Les fractions de seconde sont autorisées, avec une précision d'au moins un chiffre et un maximum illimité. Les décalages en heure locale peuvent être représentés sous forme de décalages heure:minute en UTC, ou en tant que `Z` littéral pour indiquer une heure locale en UTC. Les décalages en heure locale sont obligatoires sur les horodatages incluant l'heure et ne sont pas autorisés sur les valeurs de date.	`4:30Z' AS TIMESTAMP)`

Types Parquet pris en charge

Amazon S3 Select prend en charge les types Parquet suivants.

- `DATE`
- `DECIMAL`
- `ENUM`
- `INT(8)`
- `INT(16)`
- `INT(32)`
- `INT(64)`
- `LIST`

> **Note**
>
> Pour les sorties de type LIST Parquet, Amazon S3 Select ne prend en charge que le format JSON. Cependant, si la requête limite les données à des valeurs simples, le type LIST Parquet peut également être interrogé au format CSV.

- STRING
- Précision prise en charge par TIMESTAMP (MILLIS/MICROS/NANOS)

> **Note**
>
> Les horodatages enregistrés au format INT(96) ne sont pas pris en charge.
> En raison de la plage du type INT(64), les horodatages utilisant l'unité NANOS ne peuvent représenter que des valeurs comprises entre 1677-09-21 00:12:43 et 2262-04-11 23:47:16. Les valeurs en dehors de cette plage ne peuvent pas être représentées par l'unité NANOS.

Mappage des types Parquet aux types de données pris en charge dans Amazon S3 Select

Types Parquet	Types de données pris en charge
DATE	timestamp
DECIMAL	decimal, numeric
ENUM	string
INT(8)	int, integer
INT(16)	int, integer
INT(32)	int, integer

Types Parquet	Types de données pris en charge
INT(64)	decimal, numeric
LIST	Chaque type Parquet de la liste est mappé au type de données correspondant
STRING	string
TIMESTAMP	timestamp

Opérateurs

Amazon S3 Select prend en charge les opérateurs suivants.

Opérateurs logiques

- AND
- NOT
- OR

Opérateurs de comparaison

- <
- >
- <=
- >=
- =
- <>
- !=
- BETWEEN
- IN – Par exemple : IN ('a', 'b', 'c')

Opérateurs de correspondance de modèles

- `LIKE`
- `_` (met en correspondance n'importe quel caractère)
- `%` (met en correspondance n'importe quelle séquence de caractères)

Opérateurs unaires

- `IS NULL`
- `IS NOT NULL`

Opérateurs mathématiques

L'addition, la soustraction, la multiplication, la division et les opérateurs modulo sont pris en charge, comme suit :

- `+`
- `-`
- `*`
- `/`
- `%`

Priorité des opérateurs

Le tableau suivant indique la priorité des opérateurs par ordre décroissant.

Opérateur ou élément	Associativité	Obligatoire
-	droite	moins unaire
*, /, %	gauche	multiplication, division, opérateur modulo

Opérateur ou élément	Associativité	Obligatoire
`+, -`	gauche	addition, soustraction
`IN`		appartenance à un ensemble
`BETWEEN`		limitation à une plage
`LIKE`		mise en correspondance de modèle de chaîne
`<>`		inférieur à, supérieur à
`=`	droite	égalité, affectation
`NOT`	droite	négation logique
`AND`	gauche	conjonction logique
`OR`	gauche	disjonction logique

Mots-clés réservés

Voici la liste des mots-clés réservés pour Amazon S3 Select. Ces mot-clés comprennent des noms de fonctions, des types de données, des opérateurs, etc. nécessaires pour exécuter les expressions SQL utilisées pour interroger le contenu d'un objet.

```
absolute
action
add
all
allocate
alter
and
any
are
as
asc
assertion
at
authorization
avg
bag
begin
between
bit
bit_length
blob
bool
boolean
both
by
cascade
cascaded
case
cast
catalog
char
char_length
character
character_length
check
clob
close
coalesce
collate
collation
column
commit
connect
```

```
connection
constraint
constraints
continue
convert
corresponding
count
create
cross
current
current_date
current_time
current_timestamp
current_user
cursor
date
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
domain
double
drop
else
end
end-exec
escape
except
exception
exec
execute
exists
external
```

```
extract
false
fetch
first
float
for
foreign
found
from
full
get
global
go
goto
grant
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
into
is
isolation
join
key
language
last
leading
left
level
like
limit
list
```

```
local
lower
match
max
min
minute
missing
module
month
names
national
natural
nchar
next
no
not
null
nullif
numeric
octet_length
of
on
only
open
option
or
order
outer
output
overlaps
pad
partial
pivot
position
precision
prepare
preserve
primary
prior
privileges
procedure
public
read
real
```

```
references
relative
restrict
revoke
right
rollback
rows
schema
scroll
second
section
select
session
session_user
set
sexp
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
string
struct
substring
sum
symbol
system_user
table
temporary
then
time
timestamp
timezone_hour
timezone_minute
to
trailing
transaction
translate
translation
trim
true
```

```
tuple
union
unique
unknown
unpivot
update
upper
usage
user
using
value
values
varchar
varying
view
when
whenever
where
with
work
write
year
zone
```

Fonctions SQL

Amazon S3 Select prend en charge les fonctions SQL suivantes.

Rubriques

- Fonctions d'agrégation
- Fonctions conditionnelles
- Fonctions de conversion
- Fonctions de date
- Fonctions de chaîne

Fonctions d'agrégation

Amazon S3 Select prend en charge les fonctions d'agrégation suivantes.

Fonction	Type d'argument	Type de retour
AVG(*expressio n*)	INT, FLOAT, DECIMAL	DECIMAL pour un argument INT, FLOAT pour un argument de virgule flottante ; sinon, la même chose que le type de données d'argument.
COUNT	-	INT
MAX(*expressio n*)	INT, DECIMAL	Identique au type d'argumen ts.
MIN(*expressio n*)	INT, DECIMAL	Identique au type d'argumen ts.
SUM(*expressio n*)	INT, FLOAT, DOUBLE, DECIMAL	INT pour un argument INT, FLOAT pour un argument de virgule flottante ; sinon, la même chose que le type de données d'argument.

Exemple de SUM

Pour agréger la taille totale des objets d'un dossier dans un rapport S3 Inventory, utilisez une expression SUM.

Le rapport S3 Inventory suivant est un fichier CSV compressé avec GZIP. Il comprend trois colonnes.

- La première colonne est le nom du compartiment S3 (*DOC-EXAMPLE-BUCKET*) auquel le rapport S3 Inventory est destiné.
- La deuxième colonne est le nom de clé d'objet qui identifie de façon unique l'objet dans le compartiment.

  La valeur *example-folder/* de la première ligne correspond au dossier *example-folder*. Dans Amazon S3, lorsque vous créez un dossier dans votre compartiment, S3 crée un objet de 0 octet dont la clé est définie par le nom de dossier que vous avez fourni.

  La valeur *example-folder/object1* de la deuxième ligne correspond à l'objet *object1* du dossier *example-folder*.

  La valeur *example-folder/object2* de la troisième ligne correspond à l'objet *object2* du dossier *example-folder*.

  Pour plus d'informations sur les dossiers S3, consultez Organisation des objets dans la console Amazon S3 à l'aide de dossiers.
- La troisième colonne renvoie la taille des objets en octets.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Pour utiliser une expression SUM afin de calculer la taille totale du dossier *example-folder*, exécutez la requête SQL avec Amazon S3 Select.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 !=
 'example-folder/';
```

Résultat de la requête :

```
3581291
```

Fonctions conditionnelles

Amazon S3 Select prend en charge les fonctions conditionnelles suivantes.

Rubriques

- CASE
- COALESCE
- NULLIF

CASE

L'expression `CASE` est une expression conditionnelle similaire aux instructions `if/then/else` trouvées dans d'autres langues. `CASE` est utilisé pour spécifier un résultat lorsqu'il y a plusieurs conditions. Il existe deux types d'expressions `CASE` : simple et recherchée.

Dans les expressions `CASE` simples, une expression est comparée à une valeur. Lorsqu'une correspondance est trouvée, l'action spécifiée dans la clause `THEN` est appliquée. Si aucune correspondance n'est trouvée, l'action de la clause `ELSE` est appliquée.

Dans les expressions `CASE` recherchées, chaque `CASE` est évaluée en fonction d'une expression booléenne et l'instruction `CASE` renvoie la première expression `CASE` correspondante. Si aucune correspondance `CASE` n'est trouvée parmi les clauses `WHEN`, l'action contenue dans la clause `ELSE` est renvoyée.

Syntaxe

> **Note**
>
> Actuellement, Amazon S3 Select ne prend pas en charge `ORDER BY` ou les requêtes contenant de nouvelles lignes. Veillez à utiliser des requêtes sans saut de ligne.

Voici une instruction `CASE` simple qui est utilisée pour faire correspondre les conditions :

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Voici une instruction `CASE` recherchée utilisée pour évaluer chaque condition :

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Exemples

> **Note**
>
> Si vous utilisez la console Amazon S3 pour exécuter les exemples suivants et que votre fichier CSV contient une ligne d'en-tête, sélectionnez Exclure la première ligne de données CSV.

Exemple 1 : utilisez une simple expression CASE pour remplacer `New York City` par `Big Apple` dans une requête. Remplacer tous les autres noms de villes par `other`.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END
 FROM S3Object;
```

Résultat de la requête :

```
venuecity        |   case
-----------------+-----------
Los Angeles      | other
New York City    | Big Apple
San Francisco    | other
Baltimore        | other
...
```

Exemple 2 : utilisez une expression CASE recherchée pour affecter des numéros de groupes basés sur la valeur `pricepaid` pour les ventes de billets individuels :

```
SELECT pricepaid, CASE WHEN CAST(pricepaid as FLOAT) < 10000 THEN 'group 1' WHEN
 CAST(pricepaid as FLOAT) > 10000 THEN 'group 2' ELSE 'group 3' END FROM S3Object;
```

Résultat de la requête :

```
pricepaid |  case
-----------+---------
12624.00 | group 2
10000.00 | group 3
```

```
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

COALESCE

COALESCE évalue les arguments dans l'ordre et renvoie la première valeur non inconnue, c'est-à-dire la première valeur non nulle ou non manquante. Cette fonction ne propage pas les valeurs nulles ou manquantes.

Syntaxe

```
COALESCE ( expression, expression, ... )
```

Paramètres

expression

Expression cible sur laquelle la fonction opère.

Exemples

```
COALESCE(1)               -- 1
COALESCE(null)            -- null
COALESCE(null, null)      -- null
COALESCE(missing)         -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)         -- 1
COALESCE(null, null, 1)   -- 1
COALESCE(null, 'string')  -- 'string'
COALESCE(missing, 1)      -- 1
```

NULLIF

Pour deux expressions données, NULLIF renvoie NULL si les deux expressions ont la même valeur. Sinon, NULLIF renvoie le résultat de l'évaluation de la première expression.

Syntaxe

```
NULLIF ( expression1, expression2 )
```

Paramètres

expression1, *expression2*

Expressions cible sur lesquelles la fonction opère.

Exemples

```
NULLIF(1, 1)             -- null
NULLIF(1, 2)             -- 1
NULLIF(1.0, 1)           -- null
NULLIF(1, '1')           -- 1
NULLIF([1], [1])         -- null
NULLIF(1, NULL)          -- 1
NULLIF(NULL, 1)          -- null
NULLIF(null, null)       -- null
NULLIF(missing, null)    -- null
NULLIF(missing, missing) -- null
```

Fonctions de conversion

Amazon S3 Select prend en charge la fonction conversationnelle suivante.

Rubriques

- CAST

CAST

La fonction CAST convertit une entité, comme une expression analysée comme valeur unique, d'un type à un autre.

Syntaxe

```
CAST ( expression AS data_type )
```

Paramètres

expression

Combinaison d'un(e) ou de plusieurs valeurs, opérateurs ou fonctions SQL qui correspondent à une valeur.

data_type

Type de données cible, comme INT, dans lequel imbriquer l'expression. Pour afficher la liste des types de données pris en charge, consultez Types de données.

Exemples

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Fonctions de date

Amazon S3 Select prend en charge les fonctions de date suivantes.

Rubriques

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

DATE_ADD

À partir d'une partie de date, d'une quantité et d'un horodatage donnés, DATE_ADD renvoie un horodatage mis à jour en remplaçant la partie de date par la quantité.

Syntaxe

```
DATE_ADD( date_part, quantity, timestamp )
```

Paramètres

date_part

Spécifie quelle partie de la date modifier. Il peut s'agir de l'une des parties suivantes :

- year

- month
- day
- hour
- minute
- second

quantity

Valeur à appliquer à l'horodatage mis à jour. Les valeurs positives de *quantity* s'ajoutent à l'élément date_part de l'horodatage et les valeurs négatives se soustraient.

timestamp

Horodatage cible sur lequel la fonction opère.

Exemples

```
DATE_ADD(year, 5, `2010-01-01T`)                -- 2015-01-01 (equivalent to
 2015-01-01T)
DATE_ADD(month, 1, `2010T`)                     -- 2010-02T (result will add precision
 as necessary)
DATE_ADD(month, 13, `2010T`)                    -- 2011-02T
DATE_ADD(day, -1, `2017-01-10T`)                -- 2017-01-09 (equivalent to
 2017-01-09T)
DATE_ADD(hour, 1, `2017T`)                      -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)          -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

À partir d'une partie de date et de deux horodatages valides donnés, DATE_DIFF renvoie la différence entre les parties de date. La valeur renvoyée est un entier négatif lorsque la valeur *date_part* de *timestamp1* est supérieure à la valeur *date_part* de *timestamp2*. La valeur renvoyée est un entier positif lorsque la valeur *date_part* de *timestamp1* est inférieure à la valeur *date_part* de *timestamp2*.

Syntaxe

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Paramètres

date_part

Spécifie la partie des horodatages à comparer. Pour obtenir la définition de `date_part`, consultez DATE_ADD.

timestamp1

Premier horodatage à comparer.

timestamp2

Deuxième horodatage à comparer.

Exemples

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)           -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                    -- 4 (2010T is equivalent to
 2010-01-01T00:00:00.000Z)
DATE_DIFF(month, `2010T`, `2011T`)                      -- 12
DATE_DIFF(month, `2011T`, `2010T`)                      -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h
 apart to be 1 day apart)
```

EXTRACT

À partir d'une partie de date et d'un horodatage donnés, `EXTRACT` renvoie la valeur de la partie de date de l'horodatage.

Syntaxe

```
EXTRACT( date_part FROM timestamp )
```

Paramètres

date_part

Spécifie la partie des horodatages à extraire. Il peut s'agir de l'une des parties suivantes :

- `YEAR`
- `MONTH`

- DAY
- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

timestamp

Horodatage cible sur lequel la fonction opère.

Exemples

```
EXTRACT(YEAR FROM `2010-01-01T`)                              -- 2010
EXTRACT(MONTH FROM `2010T`)                                   -- 1 (equivalent to
 2010-01-01T00:00:00.000Z)
EXTRACT(MONTH FROM `2010-10T`)                                -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`)                -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`)              -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`)       -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`)     -- 8
```

TO_STRING

À partir d'un horodatage et d'un modèle de format donnés, TO_STRING renvoie une représentation sous forme de chaîne de l'horodatage au format donné.

Syntaxe

```
TO_STRING ( timestamp time_format_pattern )
```

Paramètres

timestamp

Horodatage cible sur lequel la fonction opère.

time_format_pattern

Chaîne qui possède les interprétations de caractères spéciaux suivantes :

Format	Exemple	Description
yy	69	Année sur deux chiffres
y	1969	Année sur quatre chiffres
yyyy	1969	Année sur 4 chiffres avec ajout de zéro
M	1	Mois de l'année
MM	01	Mois de l'année avec ajout de zéro
MMM	Jan	Nom du mois sous forme abrégée
MMMM	January	Nom du mois en entier
MMMMM	J	Première lettre du mois (REMARQUE : ce format n'est pas valide avec la fonction TO_TIMESTAMP .)
d	2	Jour du mois (1-31)

Format	Exemple	Description
dd	02	Jour du mois avec ajout de zéro (01-31)
a	AM	AM ou PM
h	3	Heure (1-12)
hh	03	Heure avec ajout de zéro (01-12)
H	3	Heure (0-23)
HH	03	Heure avec ajout de zéro (00-23)
m	4	Minutes (0-59)
mm	04	Minute avec ajout de zéro (00-59)
s	5	Secondes (0-59)
ss	05	Seconde avec ajout de zéro (00-59)
S	0	Fraction de seconde (précision : 0,1, plage : 0,0-0,9)

Format	Exemple	Description
SS	6	Fraction de seconde (précision : 0,01, plage : 0,0-0,99)
SSS	60	Fraction de seconde (précision : 0,001, plage : 0,0-0,999)
…	…	…
SSSSSSSSS	60000000	Fraction de seconde (précision maximale : 1 nanosecon de, plage : 0,0-0,999 999999)
n	60000000	Nanoseconde
X	+07 ou Z	Décalage en heures ou Z si le décalage est de 0
XX ou XXXX	+0700 ou Z	Décalage en heures et minutes ou Z si le décalage est de 0

Format	Exemple	Description
XXX ou XXXXX	+07:00 ou Z	Décalage en heures et minutes ou Z si le décalage est de 0
x	7	Décalage en heures
xx ou xxxx	700	Décalage en heures et minutes
xxx ou xxxxx	+07:00	Décalage en heures et minutes

Exemples

```
TO_STRING(`1969-07-20T20:18Z`,  'MMMM d, y')                    -- "July 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMM d, yyyy')                   -- "Jul 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'M-d-yy')                        -- "7-20-69"
TO_STRING(`1969-07-20T20:18Z`, 'MM-d-y')                        -- "07-20-1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y h:m a')               -- "July 20, 1969 8:18
 PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX')           --
 "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX')     --
 "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX')   --
 "1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX')  --
 "1969-07-20T20:18:00+08:00"
```

TO_TIMESTAMP

À partir d'une chaîne donnée, `TO_TIMESTAMP` le convertit en un horodatage. `TO_TIMESTAMP` est l'opération inverse de `TO_STRING`.

Syntaxe

```
TO_TIMESTAMP ( string )
```

Paramètres

string

Chaîne cible sur laquelle la fonction opère.

Exemples

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

`UTCNOW` renvoie l'heure actuelle au format UTC sous forme d'horodatage.

Syntaxe

```
UTCNOW()
```

Paramètres

`UTCNOW` n'accepte aucun paramètre.

Exemples

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Fonctions de chaîne

Amazon S3 Select prend en charge les fonctions de chaîne suivantes.

Rubriques

- CHAR_LENGTH, CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM

- UPPER

CHAR_LENGTH, CHARACTER_LENGTH

CHAR_LENGTH (ou CHARACTER_LENGTH) compte le nombre de caractères dans la chaîne spécifiée.

> **Note**
>
> CHAR_LENGTH et CHARACTER_LENGTH sont synonymes.

Syntaxe

```
CHAR_LENGTH ( string )
```

Paramètres

string

Chaîne cible sur laquelle la fonction opère.

Exemples

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')   -- 7
```

LOWER

À partir d'une chaîne donnée, LOWER convertit tous les caractères majuscules en minuscules. Les caractères qui ne sont pas en majuscules restent inchangés.

Syntaxe

```
LOWER ( string )
```

Paramètres

string

Chaîne cible sur laquelle la fonction opère.

Exemples

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

SUBSTRING

À partir d'une chaîne, d'un index de début et éventuellement d'une longueur donnés, `SUBSTRING` renvoie la sous-chaîne de l'index de début à la fin de la chaîne ou à la fin de la longueur précisée.

> **Note**
>
> Le premier caractère de la chaîne d'entrée possède la position d'index 1.
>
> - Si `start` est < 1, sans longueur spécifiée, la position d'index est définie sur 1.
> - Si `start` est < 1, avec une longueur spécifiée, la position d'index est définie sur `start + length -1`.
> - Si `start + length -1` est < 0, une chaîne vide est renvoyée.
> - Si `start + length -1` est > = 0, la sous-chaîne commençant à la position d'index 1 dont la longueur est `start + length - 1` est renvoyée.

Syntaxe

```
SUBSTRING( string FROM start [ FOR length ] )
```

Paramètres

`string`

Chaîne cible sur laquelle la fonction opère.

`start`

Position de début de la chaîne.

`length`

Longueur de la sous-chaîne à renvoyer. En cas d'absence du paramètre, le traitement s'effectue jusqu'à la fin de la chaîne.

Exemples

```
SUBSTRING("123456789", 0)      -- "123456789"
SUBSTRING("123456789", 1)      -- "123456789"
SUBSTRING("123456789", 2)      -- "23456789"
SUBSTRING("123456789", -4)     -- "123456789"
SUBSTRING("123456789", 0, 999) -- "123456789"
SUBSTRING("123456789", 1, 5)   -- "12345"
```

TRIM

Supprime les caractères de tête ou de fin d'une chaîne. Le caractère par défaut à supprimer est un espace (' ').

Syntaxe

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Paramètres

string

Chaîne cible sur laquelle la fonction opère.

LEADING | TRAILING | BOTH

Ce paramètre indique s'il faut supprimer les caractères de tête ou de fin ou les deux.

remove_chars

Jeu de caractères à supprimer. *remove_chars* peut être une chaîne de longueur > 1. Cette fonction renvoie la chaîne avec n'importe quel caractère de *remove_chars* se trouvant au début ou à la fin de la chaîne qui a été supprimée.

Exemples

```
TRIM('       foobar         ')              -- 'foobar'
TRIM('      \tfoobar\t         ')           -- '\tfoobar\t'
TRIM(LEADING FROM '       foobar         ')  -- 'foobar         '
TRIM(TRAILING FROM '       foobar         ') -- '       foobar'
TRIM(BOTH FROM '       foobar         ')     -- 'foobar'
```

```
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

UPPER

À partir d'une chaîne donnée, UPPER convertit tous les caractères minuscules en majuscules. Les caractères qui ne sont pas en minuscules restent inchangés.

Syntaxe

```
UPPER ( string )
```

Paramètres

string

Chaîne cible sur laquelle la fonction opère.

Exemples

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

Exécution des opérations par lot à grande échelle sur des objets Amazon S3

Vous pouvez utiliser les opérations Batch S3 pour effectuer des opérations par lot à grande échelle sur des objets Amazon S3. Les opérations par lot S3 peuvent effectuer une seule opération sur les listes d'objets Amazon S3 que vous spécifiez. Une seule tâche peut effectuer une opération spécifiée sur des milliards d'objets contenant des exaoctets de données. Amazon S3 suit la progression, envoie des notifications et stocke un rapport de fin d'opérations détaillé sur toutes les actions, offrant ainsi une expérience sans serveur entièrement gérée et contrôlable.. Vous pouvez utiliser les opérations par lot S3 via la AWS Management Console, la AWS CLI, les kits SDK Amazon ou l'API REST.

Utilisez les opérations par lot S3 pour copier des objets et définir des étiquettes d'objet ou des listes de contrôle d'accès (ACL). Vous pouvez également lancer des restaurations d'objets à partir de S3 Glacier Flexible Retrieval ou appeler une fonction AWS Lambda pour effectuer des actions personnalisées à l'aide de vos objets. Vous pouvez effectuer ces opérations sur une liste d'objets personnalisée ou utiliser un rapport Amazon S3 Inventory pour faciliter la génération de listes

d'objets. Les opérations par lot Amazon S3 utilisent les mêmes API Amazon S3 que vous utilisez déjà avec Amazon S3, de sorte que vous trouverez l'interface familière.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire. Pour plus d'informations sur l'utilisation des opérations par lots avec S3 Express One Zone et les compartiments de répertoires, consultez Utilisation des opérations par lots avec S3 Express One Zone.

Principes de base des opérations par lot S3

Vous pouvez utiliser les opérations par lot S3 pour effectuer des opérations par lot à grande échelle sur des objets Amazon S3. Les opérations par lot S3 peuvent exécuter une seule opération ou action sur des listes d'objets Amazon S3 que vous spécifiez.

Terminologie

Cette section utilise les termes tâche, opérations et sous-tâche, dont la définition est la suivante :

Tâche

Une tâche est l'unité de travail de base des opérations par lot S3. Une tâche contient toutes les informations nécessaires à l'exécution de l'opération spécifiée pour les objets répertoriés dans le manifeste. Une fois que vous avez fourni ces informations et que vous demandez le lancement de la tâche, celle-ci effectue l'opération pour chaque objet présent dans le manifeste.

Opération

L'opération est le type d'action d'API, telle que la copie d'objets, que vous souhaitez faire exécuter par la tâche d'opérations par lot Chaque tâche effectue un seul type d'opération sur tous les objets spécifiés dans le manifeste.

Sous-tâche

Une sous-tâche est l'unité d'exécution d'une tâche. Une sous-tâche représente un appel unique à une opération d'Amazon S3 ou d'API AWS Lambda en vue d'effectuer l'opération de la tâche sur un objet unique. Pendant toute la durée de vie d'une tâche, les opérations par lot S3 créent une seule tâche pour chaque objet spécifié dans le manifeste.

Fonctionnement d'une tâche d'opérations par lot S3

Une tâche est l'unité de travail de base des opérations par lot S3. Une tâche contient toutes les informations nécessaires à l'exécution de l'opération spécifiée pour une liste d'objets. Pour créer une tâche, vous fournissez aux opérations par lot S3 une liste d'objets et vous spécifiez l'action à exécuter sur ces objets.

Pour plus d'informations sur les opérations prises en charge par les opérations par lot S3, consultez Opérations prises en charge par les opérations par lot S3.

Une tâche par lot exécute une opération spécifiée sur chaque objet inclus dans son manifeste. Un manifeste répertorie les objets que vous souhaitez traiter par le biais d'une tâche par lot et il est stocké sous forme d'objet dans un compartiment. Vous pouvez utiliser un rapport Inventaire Simple Storage Service (Amazon S3) au format comma-separated values (CSV) comme manifeste, ce qui facilite la création de grandes listes d'objets situés dans un compartiment. Vous pouvez également spécifier un manifeste dans un format CSV simple qui vous permet d'effectuer des opérations par lots sur une liste personnalisée d'objets stockés dans un compartiment unique.

Une fois que vous avez créé une tâche, Amazon S3 traite la liste d'objets du manifeste et exécute l'opération spécifiée sur chaque objet. Pendant qu'une tâche est en cours, vous pouvez surveiller sa progression par programme ou via la console Amazon S3. Vous pouvez également configurer une tâche pour générer un rapport de fin lorsque celle-ci se termine. Le rapport de fin décrit les résultats de chaque sous-tâche. Pour plus d'informations sur la surveillance des tâches, consultez Gestion des tâches d'opérations par lot S3.

Tutoriel des opérations par lots S3

Le didacticiel suivant présente end-to-end des procédures complètes pour certaines tâches d'opérations par lots.

- Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3)

Avant de créer et d'exécuter des tâches d'opérations par lot S3, vous devez accorder les autorisations requises. Pour créer une tâche d'opérations par lot Amazon S3, l'autorisation de l'utilisateur `s3:CreateJob` est requise. La même entité qui crée la tâche doit également être

`iam:PassRole` autorisée à transmettre le rôle AWS Identity and Access Management (IAM) spécifié pour la tâche à Batch Operations.

Pour plus d'informations générales sur la spécification des ressources IAM, consultez Éléments de stratégie JSON IAM : Ressource dans le Guide de l'utilisateur IAM. Les sections suivantes fournissent des informations sur la création d'un rôle IAM et l'attachement de stratégies.

Rubriques

- Création d'un rôle IAM pour les opérations par lot S3
- Attachement des stratégies d'autorisations

Création d'un rôle IAM pour les opérations par lot S3

Amazon S3 doit avoir l'autorisation d'effectuer des tâches d'opérations par lot S3 en votre nom. Vous accordez ces autorisations via un rôle AWS Identity and Access Management (IAM). Cette section fournit des exemples de stratégies d'approbation et d'autorisation que vous utilisez lors de la création d'un rôle IAM. Pour de plus amples informations, veuillez consulter Rôles IAM dans le Guide de l'utilisateur IAM. Pour obtenir des exemples, veuillez consulter Contrôle des autorisations pour les opérations par lot S3 à l'aide d'étiquettes de tâche et Copie d'objets à l'aide d'opérations par lot S3.

Dans vos stratégies IAM, vous pouvez également utiliser des clés de condition pour filtrer les autorisations d'accès pour les tâches d'opérations par lot S3. Pour plus d'informations et une liste complète des clés de condition spécifiques à Amazon S3, consultez la section Actions, ressources et clés de condition pour Amazon S3 dans la référence d'autorisation de service.

Stratégie d'approbation

Vous attachez la stratégie d'approbation suivante au rôle IAM afin de permettre au principal de service de la fonctionnalité d'opérations par lot S3 d'endosser le rôle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
```

```
    ]
}
```

Attachement des stratégies d'autorisations

En fonction du type d'opérations, vous pouvez attacher l'une des stratégies suivantes.

Avant de configurer les autorisations, veuillez noter ce qui suit :

- Indépendamment de l'opération, Amazon S3 a besoin d'autorisations pour lire l'objet manifeste depuis votre compartiment S3 et, éventuellement, écrire un rapport dans votre compartiment. Dès lors, toutes les stratégies d'autorisations suivantes incluent ces autorisations.
- Pour les manifestes de rapport Amazon S3 Inventory, S3 Batch Operations nécessite l'autorisation de lire l'objet manifest.json et tous les fichiers de données CSV associés.
- Des autorisations spécifiques à la version comme `s3:GetObjectVersion` sont requises uniquement quand vous spécifiez l'ID de version des objets.
- Si vous exécutez S3 Batch Operations sur des objets chiffrés, le rôle IAM doit également avoir accès aux AWS KMS clés utilisées pour les chiffrer.
- Si vous soumettez un manifeste de rapport d'inventaire chiffré avec AWS KMS, votre politique IAM doit inclure les autorisations relatives à l'objet manifest.json `"kms:Decrypt"` et `"kms:GenerateDataKey"` à tous les fichiers de données CSV associés.
- Si la tâche Batch Operations génère un manifeste dans un bucket dans lequel les ACL sont activées et se trouve sur un autre AWS compte, vous devez accorder l'`s3:PutObjectAcl`autorisation dans la politique IAM du rôle IAM configuré pour la tâche par lots. Si vous n'incluez pas cette autorisation, le traitement par lots échoue avec l'erreur`Error occurred when preparing manifest: Failed to write manifest`.

Copier des objets : PutObject

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging"
            ],
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::DestinationBucket/*"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::SourceBucket",
                "arn:aws:s3:::SourceBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Remplacez le balisage des objets : PutObjectTagging

```
{
  "Version":"2012-10-17",
  "Statement":[
```

```
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::ReportBucket/*"
      ]
    }
  ]
}
```

Supprimer le balisage des objets : DeleteObjectTagging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "s3:DeleteObjectTagging",
              "s3:DeleteObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Remplacer la liste de contrôle d'accès : PutObjectAcl

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
      ],
      "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
```

```
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::ReportBucket/*"
      ]
    }
  ]
}
```

Restaurer les objets : RestoreObject

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "s3:RestoreObject"
      ],
      "Resource": "arn:aws:s3:::TargetResource/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::ManifestBucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
```

```
      "Resource":[
        "arn:aws:s3:::ReportBucket/*"
      ]
    }
  ]
}
```

Appliquer la rétention Object Lock : PutObjectRetention

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention",
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Appliquer la garantie légale d'Object Lock : PutObjectLegalHold

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::TargetResource"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutObjectLegalHold",
            "Resource": [
                "arn:aws:s3:::TargetResource/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::ReportBucket/*"
            ]
        }
    ]
}
```

Répliquer des objets existants : InitiateReplication avec un manifeste généré par S3

Utilisez cette politique si vous utilisez et stockez un manifeste généré par S3. Pour plus d'informations sur l'utilisation des opérations par lot pour répliquer des objets existants, consultez Réplication d'objets existants via la réplication par lot S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** manifest bucket ***/*"
         ]
```

```
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::*** completion report bucket ****/*",
            "arn:aws:s3:::*** manifest bucket ****/*"
         ]
      }
   ]
}
```

Répliquer des objets existants : InitiateReplication avec un manifeste utilisateur

Utilisez cette politique si vous utilisez un manifeste fourni par l'utilisateur. Pour plus d'informations sur l'utilisation des opérations par lot pour répliquer des objets existants, consultez Réplication d'objets existants via la réplication par lot S3.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** replication source bucket ***/*"
         ]
      },
            {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::*** manifest bucket ***/*"
         ]
      },
      {
```

```
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::*** completion report bucket ****/*"
         ]
      }
   ]
}
```

Création d'une tâche d'opérations par lot S3

Les opérations par lots Amazon S3 vous permettent d'effectuer des opérations par lots à grande échelle sur une liste d'objets Amazon S3 spécifiques. Cette section décrit les informations dont vous avez besoin pour créer une tâche d'opérations par lot S3 ainsi que les résultats d'une demande `CreateJob`. Il fournit également des instructions pour créer une tâche Batch Operations à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Lorsque vous créez une tâche d'opérations par lot S3, vous pouvez demander un rapport de fin de tâche pour toutes les tâches ou uniquement pour les tâches qui ont échoué. Tant qu'au moins une tâche a été invoquée avec succès, les opérations par lots S3 génèrent un rapport pour les tâches qui ont été achevées, qui ont échoué ou qui ont été annulées. Pour plus d'informations, consultez Exemples : Rapports de fin de tâche d'opérations par lot S3.

Rubriques

- Éléments d'une demande de tâche d'opération par lot
- Spécification d'un manifeste

Éléments d'une demande de tâche d'opération par lot

Pour créer une tâche d'opérations par lot S3, vous devez fournir les informations suivantes :

Opération

Spécifiez l'opération que la tâche d'opérations par lot S3 doit exécuter sur les objets du manifeste. Chaque type d'opération accepte des paramètres spécifiques à cette opération. Avec Batch Operations, vous pouvez effectuer une opération en bloc, avec les mêmes résultats que si vous aviez effectué cette opération one-by-one sur chaque objet.

Manifeste

Le manifeste est la liste de tous les objets sur lesquels vous voulez que les opérations par lots S3 exécutent l'opération spécifiée. Vous pouvez utiliser les méthodes suivantes pour spécifier un manifeste pour une tâche d'opérations par lot :

- Créez manuellement votre propre liste d'objets personnalisée au format CSV.
- Choisissez un rapport Inventaire Simple Storage Service (Amazon S3) existant au format CSV.
- Indiquez aux opérations par lots de générer automatiquement un manifeste en fonction des critères de filtre d'objet que vous spécifiez lors de la création de votre tâche. Cette option est disponible pour les tâches de réplication par lots que vous créez dans la console Amazon S3, ou pour tout type de tâche que vous créez à l' AWS CLI aide des AWS SDK ou de l'API REST Amazon S3.

> **Note**
>
> - Quelle que soit la manière dont vous spécifiez votre manifeste, la liste elle-même doit être stockée dans un compartiment à usage général. Les opérations par lots ne peuvent pas importer de manifestes existants ni enregistrer les manifestes générés dans des compartiments de répertoires. Toutefois, les objets décrits dans le manifeste peuvent être stockés dans des compartiments de répertoires. Pour plus d'informations, consultez Compartiments de répertoires.
> - Si les objets de votre manifeste se trouvent dans un compartiment avec gestion des versions, le fait de spécifier les ID de version des objets indique aux opérations par lots d'effectuer l'opération sur une version spécifique. Si aucun ID de version n'est spécifié, les opérations par lots effectuent l'opération sur la version la plus récente des objets. Si votre manifeste comprend un champ d'identification de version, vous devez fournir un identifiant de version pour tous les objets du manifeste.

Pour plus d'informations, consultez Spécification d'un manifeste.

Priority

Utilisez les priorités pour indiquer la priorité relative de cette tâche par rapport aux autres s'exécutant dans votre compte. Un nombre plus élevé indique une priorité plus élevée.

Les priorités n'ont de sens que par rapport aux priorités établies pour d'autres tâches dans le même compte et la même Région. Ainsi, vous pouvez choisir le système de numérotation

qui vous convient. Par exemple, vous pouvez affecter à toutes les tâches Restaurer (`RestoreObject`) la priorité 1, à toutes les tâches Copier (`CopyObject`) la priorité 2 et à toutes les tâches Remplacer une liste de contrôle d'accès (ACL) (`PutObjectAcl`) la priorité 3.

Les opérations par lots S3 traitent les tâches en fonction de leur numéro de priorité, mais un ordre strict n'est pas garanti. Par conséquent, n'utilisez pas les priorités de tâche pour vous assurer qu'une tâche commence ou finit avant une autre. Si vous devez garantir un ordre strict, attendez qu'une tâche se termine avant de démarrer la suivante.

RoleArn

Spécifiez un rôle AWS Identity and Access Management (IAM) pour exécuter la tâche. Le rôle IAM que vous utilisez doit avoir les autorisations suffisantes pour exécuter l'opération spécifiée dans la tâche. Par exemple, pour exécuter une tâche `CopyObject`, le rôle IAM doit avoir l'autorisation `s3:GetObject` pour le compartiment source et l'autorisation `s3:PutObject` pour le compartiment de destination. Le rôle a également besoin des autorisations pour lire le manifeste et écrire le rapport de fin de tâche.

Pour plus d'informations sur les rôles IAM, consultez Rôles IAM dans le manuel IAM Guide de l'utilisateur.

Pour plus d'informations sur les autorisations Amazon S3, consultez Actions politiques pour Amazon S3.

> **Note**
>
> Les tâches d'opérations par lots qui effectuent des actions sur des compartiments de répertoires nécessitent des autorisations spécifiques. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

Rapport

Indiquez si vous souhaitez que la fonctionnalité d'opérations par lot S3 génère un rapport d'achèvement. Si vous demandez un rapport de fin de tâche, vous devez également fournir les paramètres du rapport dans cet élément. Les informations nécessaires sont les suivantes :

- Le compartiment dans lequel vous souhaitez stocker le rapport

> **Note**
>
> Le rapport doit être stocké dans un compartiment à usage général. Les opérations par lots ne peuvent pas enregistrer les rapports dans des compartiments de répertoires. Pour plus d'informations, consultez Compartiments de répertoires.

- Le format du rapport
- Si vous souhaitez que le rapport comprenne les détails de toutes les tâches ou uniquement de celles ayant échoué
- Une chaîne de préfixe facultative

> **Note**
>
> Les rapports d'achèvement sont toujours chiffrés avec des clés gérées par Amazon S3 (SSE-S3).

Les balises (facultatif)

Vous pouvez étiqueter et contrôler l'accès à vos tâches d'opérations par lot S3 en ajoutant des étiquettes. Vous pouvez utiliser des balises pour identifier qui est responsable d'une tâche d'opérations par lots ou pour contrôler comment les utilisateurs interagissent avec les tâches d'opérations par lot. La présence d'étiquettes de tâche peut octroyer ou limiter la capacité d'un utilisateur à annuler une tâche, activer une tâche dans l'état de confirmation ou modifier le niveau de priorité d'une tâche. Par exemple, vous pouvez accorder à un utilisateur l'autorisation d'invoquer l'opération `CreateJob`, à condition que la tâche soit créée avec la balise `"Department=Finance"`.

Vous pouvez créer des tâches avec des étiquettes qui leur sont attachées et ajouter des étiquettes aux travaux après les avoir créés.

Pour plus d'informations, consultez the section called "Utilisation d'étiquettes".

Description (facultative)

Pour suivre et surveiller votre tâche, vous pouvez également fournir une description de 256 caractères maximum. Amazon S3 inclut cette description chaque fois qu'il renvoie des informations sur une tâche ou affiche les détails de la tâche sur la console Amazon S3. Cela vous

permet de trier et filtrer facilement les tâches en fonction des descriptions que vous avez allouées. Comme les descriptions n'ont pas à être uniques, vous pouvez utiliser les descriptions comme catégories (par exemple, « Tâches hebdomadaires de copie des journaux ») pour vous aider à suivre les groupes de tâches similaires.

Spécification d'un manifeste

Un manifeste est un objet Amazon S3 qui contient les clés d'objet sur lesquelles Amazon S3 doit agir. Vous pouvez fournir un manifeste de différentes manières :

- Créez manuellement un nouveau fichier manifeste.
- Utilisez un manifeste existant.
- Indiquez aux opérations par lots de générer automatiquement un manifeste en fonction des critères de filtre d'objet que vous spécifiez lors de la création de votre tâche. Cette option est disponible pour les tâches de réplication par lots que vous créez dans la console Amazon S3, ou pour tout type de tâche que vous créez à l' AWS CLI aide des AWS SDK ou de l'API REST Amazon S3.

> **Note**
>
> Quelle que soit la manière dont vous spécifiez votre manifeste, la liste elle-même doit être stockée dans un compartiment à usage général. Les opérations par lots ne peuvent pas importer de manifestes existants ni enregistrer les manifestes générés dans des compartiments de répertoires. Toutefois, les objets décrits dans le manifeste peuvent être stockés dans des compartiments de répertoires. Pour plus d'informations, consultez Compartiments de répertoires.

Création d'un fichier manifeste

Pour créer manuellement un fichier manifeste, vous devez spécifier la clé d'objet du manifeste, l'ETag (balise d'entité) et l'ID de version facultatif dans une liste au format CSV. Les contenus du manifeste doivent être codés au format URL.

Par défaut, Amazon S3 utilise automatiquement le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) pour chiffrer un manifeste chargé dans un compartiment Amazon S3. Les manifestes qui utilisent un chiffrement côté serveur avec des clés fournies par le client (SSE-C) ne

sont pas pris en charge. Les manifestes qui utilisent le chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ne sont pris en charge que lorsque vous utilisez des rapports d'inventaire au format CSV. L'utilisation d'un manifeste créé manuellement avec n' AWS KMS est pas prise en charge.

Votre manifeste doit contenir le nom du compartiment, la clé d'objet et, facultativement, la version de chaque objet. Les autres champs du manifeste ne sont pas utilisés lors des opérations par lot S3.

> **Note**
>
> Si les objets de votre manifeste se trouvent dans un compartiment avec gestion des versions, le fait de spécifier les ID de version des objets indique aux opérations par lots d'effectuer l'opération sur une version spécifique. Si aucun ID de version n'est spécifié, les opérations par lots effectuent l'opération sur la version la plus récente des objets. Si votre manifeste comprend un champ d'identification de version, vous devez fournir un identifiant de version pour tous les objets du manifeste.

Voici un exemple de manifeste au format CSV sans ID de version.

```
Examplebucket,objectkey1
Examplebucket,objectkey2
Examplebucket,objectkey3
Examplebucket,photos/jpgs/objectkey4
Examplebucket,photos/jpgs/newjersey/objectkey5
Examplebucket,object%20key%20with%20spaces
```

Voici un exemple de manifeste au format CSV incluant les ID de version.

```
Examplebucket,objectkey1,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
Examplebucket,objectkey2,YY_ouuAJByNW1LRBfFMfxMge7XQWxMBF
Examplebucket,objectkey3,jbo9_jhdPEyB4RrmOxWS0kU0EoNrU_oI
Examplebucket,photos/jpgs/objectkey4,6EqlikJJxLTsHsnbZbSRffn24_eh5Ny4
Examplebucket,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8GOu.NHunHO1gVs
Examplebucket,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5ArQAo3w
```

Spécification d'un fichier manifeste existant

Vous pouvez spécifier un fichier manifeste pour une demande de création de tâche dans l'un des deux formats suivants :

- Rapport d'inventaire Amazon S3 : il doit s'agir d'un rapport d'inventaire Amazon S3 au format CSV. Vous devez spécifier le fichier `manifest.json` associé au rapport d'inventaire. Pour plus d'informations sur les rapports d'inventaire, consultez Inventaire Simple Storage Service (Amazon S3). Si le rapport d'inventaire inclut des ID de version, les opérations par lot S3 s'exécutent sur les versions d'objets spécifiques.

  > **Note**
  >
  > - Les opérations par lots S3 prennent en charge des rapports d'inventaire CSV chiffrés avec SSE-KMS.
  > - Si vous soumettez un manifeste de rapport d'inventaire chiffré avec SSE-KMS, votre politique IAM doit inclure les autorisations `"kms:Decrypt"` et `"kms:GenerateDataKey"` pour l'objet `manifest.json` et tous les fichiers de données CSV associés.

- Fichier CSV : chaque ligne du fichier doit inclure le nom du compartiment, la clé d'objet et, éventuellement, la version de l'objet. Les clés d'objets doivent être encodées en URL, comme montré dans les exemples suivants. Le manifeste doit inclure soit des ID de version pour tous les objets ou omettre les ID de version pour tous les objets. Pour plus d'informations sur le format de manifeste CSV, consultez JobManifestSpec dans la Référence d'API Amazon Simple Storage Service.

  > **Note**
  >
  > Les opérations par lots S3 ne prennent pas en charge les fichiers manifestes CSV chiffrés avec SSE-KMS.

> **Important**
>
> Lorsque vous utilisez un manifeste créé manuellement et un compartiment avec gestion des versions, nous vous recommandons de spécifier les ID de version des objets. Lorsque vous créez une tâche, les opérations par lot S3 analysent l'intégralité du manifeste avant d'exécuter la tâche. Cependant, cela ne crée pas d'instantané de l'état de ce compartiment. Les manifestes pouvant contenir des milliards d'objets, l'exécution des tâches peut prendre beaucoup de temps, ce qui peut avoir une incidence sur la version d'un objet sur laquelle la tâche agit. Supposons que vous remplaciez un objet par une nouvelle version pendant

qu'une tâche s'exécute et que vous n'avez pas spécifié d'ID de version pour cet objet. Dans ce cas, Amazon S3 effectue l'opération sur la dernière version de l'objet, et non pas sur la version qui existait lorsque vous avez créé la tâche. Le seul moyen d'éviter ce comportement consiste à spécifier des ID de version pour les objets répertoriés dans le manifeste.

Génération automatique d'un manifeste

Vous pouvez indiquer à Amazon S3 de générer automatiquement un manifeste en fonction des critères de filtre d'objet que vous spécifiez lors de la création de la tâche. Cette option est disponible pour les tâches de réplication par lots que vous créez dans la console Amazon S3, ou pour tout type de tâche que vous créez à l' AWS CLI aide des AWS SDK ou de l'API REST Amazon S3. Pour en savoir plus sur la réplication par lot, consultez Réplication d'objets existants via la réplication par lot S3.

Pour générer automatiquement un manifeste, vous devez spécifier les éléments suivants dans le cadre de votre demande de création de tâche :

- Des informations sur le compartiment qui contient vos objets sources, y compris le propriétaire du compartiment et l'Amazon Resource Name (ARN)
- Des informations relatives à la sortie du manifeste, notamment un indicateur permettant de créer un fichier manifeste, le propriétaire du compartiment de sortie, l'ARN, le préfixe, le format de fichier et le type de chiffrement
- Des critères facultatifs pour filtrer les objets par date de création, nom de clé, taille, classe de stockage et balises

Critères de filtre d'objet

Pour filtrer la liste des objets à inclure dans un manifeste généré automatiquement, vous pouvez spécifier les critères suivants. Pour plus d'informations, consultez JobManifestGeneratorFilter dans la Référence d'API Amazon S3.

CreatedAfter

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source créés après cet instant.

CreatedBefore

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source créés avant cet instant.

EligibleForReplication

Si ce critère est fourni, le manifeste généré inclut les objets uniquement s’ils sont éligibles à la réplication conformément à la configuration de réplication du compartiment source.

KeyNameConstraint

S'il est fourni, le manifeste généré inclut uniquement les objets du compartiment source dont les clés d'objet correspondent aux contraintes de chaîne spécifiées pour MatchAnySubstringMatchAnyPrefix, et MatchAnySuffix.

MatchAnySubstring— S'il est fourni, le manifeste généré inclut des objets si la chaîne spécifiée apparaît n'importe où dans la chaîne clé de l'objet.

MatchAnyPrefix— S'il est fourni, le manifeste généré inclut des objets si la chaîne spécifiée apparaît au début de la chaîne clé de l'objet.

MatchAnySuffix— S'il est fourni, le manifeste généré inclut des objets si la chaîne spécifiée apparaît à la fin de la chaîne clé de l'objet.

MatchAnyStorageClass

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source stockés avec la classe de stockage spécifiée.

ObjectReplicationStatuses

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source présentant l’un des statuts de réplication spécifiés.

ObjectSizeGreaterThanBytes

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source dont la taille de fichier est supérieure au nombre d’octets spécifié.

ObjectSizeLessThanBytes

Si ce critère est fourni, le manifeste généré inclut uniquement les objets du compartiment source dont la taille de fichier est inférieure au nombre d’octets spécifié.

> **Note**
>
> Vous ne pouvez pas cloner la plupart des tâches qui ont généré automatiquement des manifestes. Les tâches de réplication par lot peuvent être clonées, sauf lorsqu'elles utilisent les critères de filtre de manifeste `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` ou `ObjectSizeLessThanBytes`.

La syntaxe permettant de spécifier les critères du manifeste varie en fonction de la méthode que vous utilisez pour créer votre tâche. Pour obtenir des exemples, consultez Création d'une tâche.

Création d'une tâche

Vous pouvez créer des tâches S3 Batch Operations à l'aide de la console Amazon S3 AWS CLI, AWS des SDK ou de l'API REST Amazon S3.

Pour plus d'informations sur la création d'une demande de tâche, consultez Éléments d'une demande de tâche d'opération par lot.

Prérequis

Avant de créer une tâche d'opérations par lot, vérifiez que vous avez configuré les autorisations pertinentes. Pour plus d'informations, consultez Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3).

Utiliser la console S3.

Pour créer une tâche par lot

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom du fichier actuellement affiché Région AWS. Choisissez ensuite la région dans laquelle vous souhaitez créer votre emploi.

   > **Note**
   >
   > Pour les opérations de copie, vous devez créer la tâche dans la même région que le compartiment de destination. Pour toutes les autres opérations, vous devez créer la tâche dans la même région que les objets du manifeste.

3. Choisissez Batch Operations dans le volet de navigation de gauche de la console Amazon S3.
4. Choisissez Créer une tâche.
5. Consultez l'Région AWSendroit où vous souhaitez créer votre emploi.
6. Sous Format du manifeste, choisissez le type d'objet manifeste à utiliser.

   - Si vous choisissez S3 inventory report (Rapport sur l'inventaire S3), entrez le chemin de l'objet manifest.json généré par Amazon S3 dans le cadre du rapport d'inventaire au format CSV et éventuellement l'ID de version de l'objet manifeste si vous souhaitez utiliser une version autre que la version la plus récente.
   - Si vous choisissez CSV, entrez le chemin vers un objet manifeste au format CSV. L'objet manifeste doit respecter le format décrit dans la console. Vous pouvez éventuellement inclure l'ID de version de l'objet manifeste si vous souhaitez utiliser une version autre que la plus récente.

   > **Note**
   >
   > La console Amazon S3 prend en charge la génération automatique de manifeste uniquement pour les tâches de réplication par lot. Pour tous les autres types de tâches, si vous souhaitez qu'Amazon S3 génère automatiquement un manifeste en fonction des critères de filtre que vous spécifiez, vous devez configurer votre tâche à l' AWS CLI aide des AWS SDK ou de l'API REST Amazon S3.

7. Choisissez Suivant.
8. Sous Operation (Opération), choisissez l'opération que vous souhaitez effectuer sur tous les objets répertoriés dans le manifeste. Remplissez les informations relatives à l'opération choisie, puis sélectionnez Suivant.
9. Remplissez les informations relatives à l'option Configure additional options (Configurer les options supplémentaires), puis sélectionnez Suivant.
10. Sous Vérification, vérifiez les paramètres. Si vous devez apporter des modifications, choisissez Précédent. Sinon, choisissez Create Job (Créer la tâche).

En utilisant le AWS CLI

Specify manifest

L’exemple suivant montre comment créer une tâche d’opérations par lots S3 `S3PutObjectTagging` qui agit sur les objets répertoriés dans un fichier manifeste existant.

Pour créer une tâche d'opérations par lot **`S3PutObjectTagging`**

1. Utilisez les commandes suivantes pour créer un rôle AWS Identity and Access Management (IAM), puis créez une politique IAM pour attribuer les autorisations appropriées. Le rôle et la politique suivants accordent à Amazon S3 l’autorisation d’ajouter des balises d’objet, dont vous aurez besoin lors de la création de la tâche dans une étape ultérieure.

   a. Utilisez l’exemple de commande suivant pour créer un rôle IAM qui sera utilisé par les opérations par lot. Pour utiliser cet exemple de commande, remplacez *`S3BatchJobRole`* par le nom que vous souhaitez donner à ce rôle.

```
aws iam create-role \
 --role-name S3BatchJobRole \
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}'
```

   Prenez note de l'Amazon Resource Name (ARN) du rôle. Vous aurez besoin de l’ARN lors de la création d’une tâche.

   b. Utilisez l’exemple de commande suivant pour créer une politique IAM avec les autorisations nécessaires et l’attacher au rôle IAM que vous avez créé à l’étape précédente. Pour plus d’informations sur les autorisations nécessaires, consultez Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3).

> **Note**
>
> Les tâches d'opérations par lots qui effectuent des actions sur des compartiments de répertoires nécessitent des autorisations spécifiques. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

Pour utiliser cet exemple de commande, remplacez les *user input placeholders* comme suit :

- Remplacez *S3BatchJobRole* par le nom de votre rôle IAM. Assurez-vous que ce nom correspond au nom que vous avez utilisé précédemment.
- Remplacez *PutObjectTaggingBatchJobPolicy* par le nom que vous souhaitez donner à la politique IAM.
- Remplacez *example-s3-destination-bucket* par le nom du compartiment qui contient les objets auxquels vous souhaitez appliquer des balises.
- Remplacez *DOC-EXAMPLE-MANIFEST-BUCKET* par le nom du compartiment qui contient le manifeste.
- Remplacez *DOC-EXAMPLE-REPORT-BUCKET* par le nom du compartiment dans lequel vous voulez livrer le rapport d'achèvement.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name PutObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket/*"
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-BUCKET/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource":[
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*"
      ]
    }
  ]
}'
```

2. Utilisez l'exemple de commande suivant pour créer une tâche `S3PutObjectTagging`.

   Le fichier `manifest.csv` fournit une liste des valeurs de compartiment et de clé d'objet. Cette tâche applique les balises spécifiées aux objets identifiés dans le manifeste. `ETag` est l'ETag de l'objet `manifest.csv` que vous pouvez obtenir à partir de la console Amazon S3. Cette demande spécifie le paramètre `no-confirmation-required`, afin que vous puissiez exécuter la tâche sans avoir à la confirmer avec la commande `update-job-status`. Pour plus d'informations, consultez la section create-job dans la référence des commandes AWS CLI .

   Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations. Remplacez *`IAM-role`* par l'ARN du rôle IAM que vous avez créé auparavant.

```
aws s3control create-job \
    --region us-west-2 \
```

```
    --account-id acct-id \
    --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"keyOne",
 "Value":"ValueOne"}] }}' \
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":
{"ObjectArn":"arn:aws:s3:::my_manifests/
manifest.csv","ETag":"60e460c9d1046e73f7dde5043ac3ae85"}}' \
    --report '{"Bucket":"arn:aws:s3:::DOC-EXAMPLE-REPORT-
BUCKET","Prefix":"final-reports",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
    --priority 42 \
    --role-arn IAM-role \
    --client-request-token $(uuidgen) \
    --description "job description" \
    --no-confirmation-required
```

En réponse, Amazon S3 renvoie un ID de tâche (par exemple, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Vous aurez besoin de l'ID de la tâche pour identifier, surveiller et modifier la tâche.

Generate manifest

L'exemple suivant montre comment créer une tâche d'opérations par lots S3 `S3DeleteObjectTagging` qui génère automatiquement un manifeste en fonction de vos critères de filtre d'objet. Ces critères incluent la date de création, le nom de la clé, la taille, la classe de stockage et les balises.

Pour créer une tâche d'opérations par lot **`S3DeleteObjectTagging`**

1. Utilisez les commandes suivantes pour créer un rôle AWS Identity and Access Management (IAM), puis créez une politique IAM pour attribuer des autorisations. Le rôle et la politique suivants accordent à Amazon S3 l'autorisation de supprimer des balises d'objet, dont vous aurez besoin lors de la création de la tâche dans une étape ultérieure.

   a. Utilisez l'exemple de commande suivant pour créer un rôle IAM qui sera utilisé par les opérations par lot. Pour utiliser cet exemple de commande, remplacez *S3BatchJobRole* par le nom que vous souhaitez donner à ce rôle.

   ```
   aws iam create-role \
    --role-name S3BatchJobRole \
   ```

```
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}'
```

Prenez note de l'Amazon Resource Name (ARN) du rôle. Vous aurez besoin de l'ARN lors de la création d'une tâche.

b. Utilisez l'exemple de commande suivant pour créer une politique IAM avec les autorisations nécessaires et l'attacher au rôle IAM que vous avez créé à l'étape précédente. Pour plus d'informations sur les autorisations nécessaires, consultez Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3).

> **Note**
>
> Les tâches d'opérations par lots qui effectuent des actions sur des compartiments de répertoires nécessitent des autorisations spécifiques. Pour plus d'informations, consultez AWS Identity and Access Management (IAM) pour S3 Express One Zone.

Pour utiliser cet exemple de commande, remplacez les *`user input placeholders`* comme suit :

- Remplacez *`S3BatchJobRole`* par le nom de votre rôle IAM. Assurez-vous que ce nom correspond au nom que vous avez utilisé précédemment.
- Remplacez *`DeleteObjectTaggingBatchJobPolicy`* par le nom que vous souhaitez donner à la politique IAM.
- Remplacez *`example-s3-destination-bucket`* par le nom du compartiment qui contient les objets auxquels vous souhaitez appliquer des balises.

- Remplacez *DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET* par le nom du compartiment dans lequel vous voulez enregistrer le manifeste.
- Remplacez *DOC-EXAMPLE-REPORT-BUCKET* par le nom du compartiment dans lequel vous voulez livrer le rapport d'achèvement.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name DeleteObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket/*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutInventoryConfiguration"
      ],
      "Resource": "arn:aws:s3:::example-s3-destination-bucket"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
```

```
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource":[
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET/*",
        "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET/*"
      ]
    }
  ]
}'
```

2. Utilisez l'exemple de commande suivant pour créer la tâche `S3DeleteObjectTagging`.

   Dans cet exemple, les valeurs de la section `--report` spécifient le compartiment, le préfixe, le format et la portée du rapport de tâche qui sera généré. La section `--manifest-generator` fournit des informations sur le compartiment source qui contient les objets sur lesquels la tâche agira, des informations sur la liste de sortie du manifeste qui sera générée pour la tâche et les critères de filtre permettant de réduire l'étendue des objets à inclure dans le manifeste en fonction de la date de création, de contraintes de nom, de la taille et de la classe de stockage. La commande spécifie également la priorité de la tâche, le rôle IAM et la Région AWS.

   Pour plus d'informations, consultez la section create-job dans la référence des commandes AWS CLI .

   Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations. Remplacez *`IAM-role`* par l'ARN du rôle IAM que vous avez créé auparavant.

```
aws s3control create-job \
    --account-id 012345678901 \
    --operation '{
        "S3DeleteObjectTagging": {}
    }' \
    --report '{
        "Bucket":"arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET",
        "Prefix":"reports",
        "Format":"Report_CSV_20180820",
        "Enabled":true,
        "ReportScope":"AllTasks"
```

```
    }' \
    --manifest-generator '{
        "S3JobManifestGenerator": {
          "ExpectedBucketOwner": "012345678901",
          "SourceBucket": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
          "EnableManifestOutput": true,
          "ManifestOutputLocation": {
            "ExpectedManifestBucketOwner": "012345678901",
            "Bucket": "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-OUTPUT-BUCKET",
            "ManifestPrefix": "prefix",
            "ManifestFormat": "S3InventoryReport_CSV_20211130"
          },
          "Filter": {
            "CreatedAfter": "2023-09-01",
            "CreatedBefore": "2023-10-01",
            "KeyNameConstraint": {
              "MatchAnyPrefix": [
                "prefix"
              ],
              "MatchAnySuffix": [
                "suffix"
              ]
            },
            "ObjectSizeGreaterThanBytes": 100,
            "ObjectSizeLessThanBytes": 200,
            "MatchAnyStorageClass": [
              "STANDARD",
              "STANDARD_IA"
            ]
          }
        }
      }' \
     --priority 2 \
     --role-arn IAM-role \
     --region us-east-1
```

En réponse, Amazon S3 renvoie un ID de tâche (par exemple, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Vous aurez besoin de cet ID de tâche pour identifier, surveiller ou modifier la tâche.

En utilisant le AWS SDK for Java

Specify manifest

L'exemple suivant montre comment créer une tâche d'opérations par lots S3 `S3PutObjectTagging` qui agit sur les objets répertoriés dans un fichier manifeste existant. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

Example

```
package aws.example.s3control;



import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.*;

import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateJob {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String iamRoleArn = "IAM Role ARN";
        String reportBucketName = "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET";
        String uuid = UUID.randomUUID().toString();

        ArrayList tagSet = new ArrayList<S3Tag>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));


        try {
            JobOperation jobOperation = new JobOperation()
                    .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                            .withTagSet(tagSet)
                    );
```

```
            JobManifest manifest = new JobManifest()
                    .withSpec(new JobManifestSpec()
                            .withFormat("S3BatchOperations_CSV_20180820")
                            .withFields(new String[]{
                                    "Bucket", "Key"
                            }))
                    .withLocation(new JobManifestLocation()
                            .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")
                            .withETag("60e460c9d1046e73f7dde5043ac3ae85"));
            JobReport jobReport = new JobReport()
                    .withBucket(reportBucketName)
                    .withPrefix("reports")
                    .withFormat("Report_CSV_20180820")
                    .withEnabled(true)
                    .withReportScope("AllTasks");

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.createJob(new CreateJobRequest()
                    .withAccountId(accountId)
                    .withOperation(jobOperation)
                    .withManifest(manifest)
                    .withReport(jobReport)
                    .withPriority(42)
                    .withRoleArn(iamRoleArn)
                    .withClientRequestToken(uuid)
                    .withDescription("job description")
                    .withConfirmationRequired(false)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Generate manifest

L'exemple suivant montre comment créer une tâche d'opérations par lots S3 `s3PutObjectCopy` qui génère automatiquement un manifeste en fonction de vos critères de filtre d'objet, dont notamment la date de création, le nom de clé et la taille. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

Example

```
    package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.CreateJobRequest;
import com.amazonaws.services.s3control.model.CreateJobResult;
import com.amazonaws.services.s3control.model.JobManifestGenerator;
import com.amazonaws.services.s3control.model.JobManifestGeneratorFilter;
import com.amazonaws.services.s3control.model.JobOperation;
import com.amazonaws.services.s3control.model.JobReport;
import com.amazonaws.services.s3control.model.KeyNameConstraint;
import com.amazonaws.services.s3control.model.S3JobManifestGenerator;
import com.amazonaws.services.s3control.model.S3ManifestOutputLocation;
import com.amazonaws.services.s3control.model.S3SetObjectTaggingOperation;
import com.amazonaws.services.s3control.model.S3Tag;

import java.time.Instant;
import java.util.Date;
import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class test {
    public static void main(String[] args) {
        String accountId = "012345678901";
        String iamRoleArn = "arn:aws:iam::012345678901:role/ROLE";
        String sourceBucketName = "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET";
```

```
            String reportBucketName = "arn:aws:s3:::DOC-EXAMPLE-REPORT-BUCKET";
            String manifestOutputBucketName = "arn:aws:s3:::DOC-EXAMPLE-MANIFEST-
OUTPUT-BUCKET";
            String uuid = UUID.randomUUID().toString();
            long minimumObjectSize = 100L;

            ArrayList<S3Tag> tagSet = new ArrayList<>();
            tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));

            ArrayList<String> prefixes = new ArrayList<>();
            prefixes.add("s3KeyStartsWith");

            try {
                JobOperation jobOperation = new JobOperation()
                        .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                                .withTagSet(tagSet)
                        );
                S3ManifestOutputLocation manifestOutputLocation = new
 S3ManifestOutputLocation()
                        .withBucket(manifestOutputBucketName)
                        .withManifestPrefix("manifests")
                        .withExpectedManifestBucketOwner(accountId)
                        .withManifestFormat("S3InventoryReport_CSV_20211130");

                JobManifestGeneratorFilter jobManifestGeneratorFilter = new
 JobManifestGeneratorFilter()
                        .withEligibleForReplication(true)
                        .withKeyNameConstraint(
                                new KeyNameConstraint()
                                        .withMatchAnyPrefix(prefixes))
                        .withCreatedBefore(Date.from(Instant.now()))
                        .withObjectSizeGreaterThanBytes(minimumObjectSize);

                S3JobManifestGenerator s3JobManifestGenerator = new
 S3JobManifestGenerator()
                        .withEnableManifestOutput(true)
                        .withManifestOutputLocation(manifestOutputLocation)
                        .withFilter(jobManifestGeneratorFilter)
                        .withSourceBucket(sourceBucketName);

                JobManifestGenerator jobManifestGenerator = new
 JobManifestGenerator()
                        .withS3JobManifestGenerator(s3JobManifestGenerator);
```

```
                JobReport jobReport = new JobReport()
                        .withBucket(reportBucketName)
                        .withPrefix("reports")
                        .withFormat("Report_CSV_20180820")
                        .withEnabled(true)
                        .withReportScope("AllTasks");

                AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                        .withCredentials(new ProfileCredentialsProvider())
                        .withRegion(US_WEST_2)
                        .build();

                CreateJobResult createJobResult = s3ControlClient.createJob(new
CreateJobRequest()
                        .withAccountId(accountId)
                        .withOperation(jobOperation)
                        .withManifestGenerator(jobManifestGenerator)
                        .withReport(jobReport)
                        .withPriority(42)
                        .withRoleArn(iamRoleArn)
                        .withClientRequestToken(uuid)
                        .withDescription("job description")
                        .withConfirmationRequired(true)
                );

                System.out.println("Created job " + createJobResult.getJobId());

            } catch (AmazonServiceException e) {
                // The call was transmitted successfully, but Amazon S3 couldn't
process
                // it and returned an error response.
                e.printStackTrace();
            } catch (SdkClientException e) {
                // Amazon S3 couldn't be contacted for a response, or the client
                // couldn't parse the response from Amazon S3.
                e.printStackTrace();
            }
        }
    }
```

Utilisation de l'API REST

Vous pouvez utiliser l'API REST pour créer une tâche d'opérations par lot. Pour plus d'informations, veuillez consulter CreateJob dans la Référence d'API Amazon Simple Storage Service.

Réponses à la tâche

Si la demande `CreateJob` aboutit, Amazon S3 renvoie un ID de tâche. L'ID de tâche est un identifiant unique généré automatiquement par Amazon S3 afin que vous puissiez identifier votre tâche d'opérations par lot et surveiller son statut.

Lorsque vous créez une tâche via les AWS CLI AWS SDK ou l'API REST, vous pouvez configurer S3 Batch Operations pour commencer à traiter la tâche automatiquement. La tâche s'exécute dès qu'elle est prête, au lieu d'attendre que des tâches de plus haute priorité soient traitées en premier.

Lorsque vous créez une tâche via la console Amazon S3, vous devez vérifier les détails de la tâche et confirmer que vous voulez l'exécuter avant que les opérations par lots commencent à la traiter. Si une tâche conserve l'état Suspendu pendant plus de 30 jours, cette tâche échouera.

Opérations prises en charge par les opérations par lot S3

Les opérations par lot S3 prennent en charge plusieurs opérations différentes. Les rubriques de cette section décrivent chacune des opérations.

Copie d'objets

L'opération Copy (Copie) copie chaque objet spécifié dans le manifeste. Vous pouvez copier des objets vers un compartiment dans la même Région AWS ou dans une Région différente. Les opérations par lot S3 supportent la plupart des options disponibles via Amazon S3 pour copier des objets. Ces options comprennent la définition des métadonnées d'objet, la spécification d'autorisations et la modification de la classe de stockage d'un objet.

Vous pouvez utiliser l'opération de copie pour copier des objets non chiffrés et les réécrire dans le même compartiment en tant qu'objets chiffrés. Pour de plus amples informations, veuillez consulter Chiffrement d'objets avec des opérations par lot Amazon S3.

Lorsque vous copiez des objets, vous pouvez modifier l'algorithme de contrôle utilisé pour calculer le total de contrôle de l'objet. Si les objets n'ont pas de total de contrôle calculé, vous pouvez également en ajouter un en spécifiant l'algorithme de total de contrôle à utiliser pour Amazon S3. Pour plus d'informations, consultez Vérification de l'intégrité des objets.

Pour de plus amples informations sur la copie d'objets dans Amazon S3, ainsi que sur les paramètres obligatoires et facultatifs, veuillez consulter la section Copier, déplacer et renommer des objets de ce guide et la section CopyObject de la Référence API du service Amazon Simple Storage.

Limites et restrictions

- Tous les objets sources doivent se trouver dans un même compartiment.
- Tous les objets de destination doivent se trouver dans un même compartiment.
- Vous devez disposer d'autorisations de lecture pour le compartiment source et d'autorisations d'écriture pour le compartiment de destination.
- La taille des objets à copier peut aller jusqu'à 5 Go.
- Si vous essayez de copier des objets depuis les classes S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive dans la classe de stockage S3 Standard, vous devez d'abord restaurer ces objets. Pour plus d'informations, consultez Restauration d'un objet archivé.
- Les tâches de copie doivent être créées dans la Région de destination, c'est-à-dire la Région dans laquelle vous prévoyez de copier les objets.
- Toutes les options de copie sont prises en charge, à l'exception des vérifications conditionnelles sur les ETags et du chiffrement côté serveur avec les clés de chiffrement fournies par le client (SSE-C).
- Si les compartiments ne sont associés à aucune version, vous remplacerez les objets dont le nom de clé est le même.
- Les objets ne sont pas nécessairement copiés dans le même ordre que celui dans lequel ils apparaissent dans le manifeste. Pour les compartiments activés pour le contrôle de version, si la préservation de l'ordre des versions actuelles ou anciennes est importante, vous devez d'abord copier toutes les anciennes versions. Ensuite, une fois la première tâche terminée, copiez les versions actuelles dans une tâche ultérieure.
- La copie d'objets vers la classe de stockage à redondance réduite (RRS) n'est pas prise en charge.

Copie d'objets à l'aide d'opérations par lot S3

Vous pouvez utiliser la fonctionnalité d'opérations par lot S3 pour créer une tâche de copie PUT dans le but de copier des objets dans un même compte ou dans un autre compte de destination. Les sections suivantes contiennent des exemples de comment stocker et utiliser un manifeste qui se trouve dans un compte différent. Dans la première section, vous pouvez utiliser l'inventaire Simple Storage Service (Amazon S3) pour livrer le rapport d'inventaire au compte de destination en vue de

son utilisation lors de la création de tâches, ou utiliser un manifeste au format CSV (séparé par des virgules) dans le compte source ou de destination comme indiqué dans le deuxième exemple. Le troisième exemple montre comment utiliser l'opération de copie pour activer le chiffrement de la clé de compartiment S3 sur des objets existants.

Exemples d'opérations de copie

- Utilisation d'un rapport d'inventaire livré au compte de destination pour copier des objets dans des Comptes AWS
- Utilisation d'un manifeste CSV stocké dans le compte source pour copier des objets dans des Comptes AWS
- Utilisation d'opérations pat lot S3 pour chiffrer des objets avec des clés de compartiment S3

Utilisation d'un rapport d'inventaire livré au compte de destination pour copier des objets dans des Comptes AWS

Vous pouvez utiliser un inventaire Amazon S3 pour créer un rapport d'inventaire, et utiliser celui-ci pour créer une liste d'objets à copier avec les opérations par lot S3 Pour de plus amples informations sur l'utilisation d'un manifeste CSV dans le compte source ou de destination, veuillez consulter the section called “Utilisation d'un manifeste CSV pour copier des objets dans des Comptes AWS”.

L'inventaire Amazon S3 génère des inventaires des objets dans un compartiment. La liste générée est publiée dans un fichier de sortie. Le compartiment qui est inventorié est appelé le compartiment source et le compartiment où le fichier de rapport d’inventaire est stocké est appelé le compartiment de destination.

Vous pouvez configurer le rapport Amazon S3 Inventory de façon à ce qu'il soit livré à un autre Compte AWS. Cela permet à la fonctionnalité d'opérations par lot S3 de lire le rapport d'inventaire lorsque la tâche a été créée dans le compte de destination.

Pour de plus amples informations sur les compartiments sources et de destination de l'inventaire Amazon S3, veuillez consulter Compartiments source et de destination.

Le moyen le plus simple de configurer un inventaire consiste à utiliser la AWS Management Console, mais vous pouvez également utiliser l'API REST, la AWS Command Line Interface (AWS CLI) ou des kits SDK AWS.

La procédure de console suivante contient les étapes de haut niveau permettant de configurer des autorisations pour une tâche d'opérations par lot S3. Dans cette procédure, vous copiez des

objets d'un compte source vers un compte de destination, avec le rapport d'inventaire stocké dans le compte de destination.

Pour configurer Amazon S3 Inventory pour des compartiments source et de destination détenus par des comptes différents

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Choisissez un compartiment de destination où stocker le rapport d'inventaire.

   Déterminez un compartiment de manifeste de destination où stocker le rapport d'inventaire. Dans cette procédure, le compte de destination est le compte qui possède à la fois le compartiment de manifeste de destination et le compartiment où les objets sont copiés.
3. Configurez un inventaire pour répertorier les objets dans un compartiment source et publier la liste dans le compartiment de manifeste de destination.

   Configurez une liste d'inventaire pour un compartiment source. Ce faisant, vous spécifiez le compartiment de destination où vous souhaitez stocker la liste. Le rapport d'inventaire pour le compartiment source est publié dans le compartiment de destination. Dans cette procédure, le compte source est le compte qui possède le compartiment source.

   Pour plus d'informations sur l'utilisation de la console pour configurer un inventaire ou sur le chiffrement d'un fichier de liste d'inventaire, consultez Configuration d'Amazon S3 Inventory.

   Choisissez CSV comme format de sortie.

   Lorsque vous saisissez des informations pour le compartiment de destination, choisissez Compartiments d'un autre compte. Ensuite, saisissez le nom du compartiment de manifeste de destination. Éventuellement, vous pouvez saisir l'ID du compte de destination.

   Une fois que la configuration d'inventaire a été enregistrée, la console affiche un message similaire au message suivant :

   Amazon S3 n'a pas pu créer une stratégie sur le compartiment de destination. Demandez au propriétaire du compartiment de destination d'ajouter la stratégie de compartiment suivante pour permettre à Amazon S3 d'ajouter des données dans ce compartiment.

   Ensuite, la console affiche une stratégie de compartiment que vous pouvez utiliser pour le compartiment de destination.

4. Copiez la stratégie de compartiment de destination qui apparaît sur la console.
5. Dans le compte de destination, ajoutez la stratégie de compartiment copiée dans le compartiment de manifeste de destination où le rapport d'inventaire est stocké.
6. Créez un rôle dans le compte de destination basé sur la stratégie d'approbation de la fonctionnalité d'opérations par lot S3. Pour de plus amples informations sur la stratégie d'approbation, veuillez consulter Stratégie d'approbation.

   Pour de plus amples informations sur la création d'un rôle, veuillez consulter la section Création d'un rôle pour la délégation d'autorisations à un service AWS du Guide de l'utilisateur IAM.

   Saisissez un nom pour le rôle (l'exemple de rôle utilise le nom `BatchOperationsDestinationRoleCOPY`). Choisissez le service S3, puis choisissez le cas d'utilisation Opérations par lot pour le compartiment S3, lequel applique la stratégie d'approbation au rôle.

   Ensuite, choisissez Create policy (Créer une stratégie) pour attacher la stratégie suivante au rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::ObjectDestinationBucket/*",
        "arn:aws:s3:::ObjectSourceBucket/*",
```

```
        "arn:aws:s3:::ObjectDestinationManifestBucket/*"
      ]
    }
  ]
}
```

Le rôle utilise la stratégie pour autoriser `batchoperations.s3.amazonaws.com` à lire le manifeste dans le compartiment de destination. Il octroie également des autorisations pour obtenir (GET) des objets, des listes de contrôle d'accès (ACL), des étiquettes et des versions dans le compartiment d'objet source. Enfin, il octroie des autorisations pour placer (PUT) des objets, des listes de contrôle d'accès (ACL), des étiquettes et des versions dans le compartiment d'objet de destination.

7. Dans le compte source, créez une stratégie de compartiment pour le compartiment source qui octroie au rôle créé à l'étape précédente l'autorisation d'obtenir (GET) des objets, des listes de contrôle d'accès, des étiquettes et des versions dans le compartiment source. Cette étape permet à la fonctionnalité d'opérations par lot S3 de récupérer des objets du compartiment source au moyen du rôle approuvé.

   Vous trouverez, ci-après, un exemple de stratégie de compartiment pour le compte source.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::DestinationAccountNumber:role/
BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::ObjectSourceBucket/*"
    }
  ]
```

```
}
```

8. Une fois le rapport d'inventaire disponible, créez une tâche d'opérations par lot S3 PUT object copy dans le compte de destination, en choisissant le rapport d'inventaire dans le compartiment de manifeste de destination. Vous avez besoin de l'ARN pour le rôle que vous avez créé dans le compte de destination.

   Pour obtenir des informations générales sur la création d'une tâche, veuillez consulter Création d'une tâche d'opérations par lot S3.

   Pour en savoir plus sur la création d'une tâche à l'aide de la console, consultez Création d'une tâche d'opérations par lot S3.

Utilisation d'un manifeste CSV stocké dans le compte source pour copier des objets dans des Comptes AWS

Vous pouvez utiliser un fichier CSV stocké dans un autre Compte AWS en tant que manifeste pour une tâche d'opérations par lot S3. Pour utiliser un rapport d'inventaire S3, veuillez consulter the section called "Utilisation d'un rapport d'inventaire pour copier des objets dans des Comptes AWS".

La procédure suivante montre comment configurer des autorisations lors de l'utilisation d'une tâche d'opérations par lot S3 pour copier des objets d'un compte source vers un compte de destination avec le fichier manifeste CSV stocké dans le compte source.

Pour configurer un manifeste CSV stocké dans un autre Compte AWS

1. Créez un rôle dans le compte de destination basé sur la stratégie d'approbation de la fonctionnalité d'opérations par lot S3. Dans cette procédure, le compte de destination est le compte dans lequel les objets sont copiés.

   Pour de plus amples informations sur la stratégie d'approbation, veuillez consulter Stratégie d'approbation.

   Pour de plus amples informations sur la création d'un rôle, veuillez consulter la section Création d'un rôle pour la délégation d'autorisations à un service AWS du Guide de l'utilisateur IAM.

   Si vous créez le rôle au moyen de la console, saisissez un nom pour le rôle (l'exemple de rôle utilise le nom `BatchOperationsDestinationRoleCOPY`). Choisissez le service S3, puis choisissez le cas d'utilisation Opérations par lot pour le compartiment S3, lequel applique la stratégie d'approbation au rôle.

Ensuite, choisissez Create policy (Créer une stratégie) pour attacher la stratégie suivante au rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::ObjectDestinationBucket/*",
        "arn:aws:s3:::ObjectSourceBucket/*",
        "arn:aws:s3:::ObjectSourceManifestBucket/*"
      ]
    }
  ]
}
```

Au moyen de la stratégie, le rôle octroie à `batchoperations.s3.amazonaws.com` l'autorisation de lire le manifeste dans le compartiment manifeste source. Il octroie des autorisations pour obtenir (GET) des objets, des listes de contrôle d'accès (ACL), des étiquettes et des versions dans le compartiment d'objet source. Enfin, il octroie des autorisations pour placer (PUT) des objets, des listes de contrôle d'accès (ACL), des étiquettes et des versions dans le compartiment d'objet de destination.

2. Dans le compte source, créez une stratégie de compartiment pour le compartiment qui contient le manifeste afin d'octroyer au rôle créé à l'étape précédente l'autorisation d'obtenir (GET) des objets et des versions dans le compartiment manifeste source.

Cette étape permet à la fonctionnalité d'opérations par lot S3 de lire le manifeste au moyen du rôle approuvé. Appliquez la stratégie de compartiment au compartiment qui contient le manifeste.

Vous trouverez, ci-après, un exemple de stratégie de compartiment à appliquer au compartiment manifeste source.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceManfiestRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::DestinationAccountNumber:user/ConsoleUserCreatingJob",
          "arn:aws:iam::DestinationAccountNumber:role/
BatchOperationsDestinationRoleCOPY"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::ObjectSourceManifestBucket/*"
    }
  ]
}
```

Cette stratégie octroie également à un utilisateur de la console qui est en train de créer une tâche dans le compte de destination les mêmes autorisations dans le compartiment manifeste source au moyen de la même stratégie de compartiment.

3. Dans le compte source, créez une stratégie de compartiment pour le compartiment source qui octroie au rôle créé l'autorisation de récupérer (paramètre GET) des objets, des listes de contrôle d'accès, des étiquettes et des versions dans le compartiment d'objet source. La fonctionnalité d'opérations par lot S3 peut ensuite récupérer des objets du compartiment source via le rôle approuvé.

   Vous trouverez, ci-après, un exemple de stratégie de compartiment pour le compartiment qui contient les objets source.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::DestinationAccountNumber:role/
BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::ObjectSourceBucket/*"
    }
  ]
}
```

4. Créez une tâche d'opérations par lot S3 dans le compte de destination. Vous avez besoin de l'ARN (Amazon Resource Name) pour le rôle que vous avez créé dans le compte de destination.

   Pour obtenir des informations générales sur la création d'une tâche, veuillez consulter Création d'une tâche d'opérations par lot S3.

   Pour en savoir plus sur la création d'une tâche à l'aide de la console, veuillez consulter Création d'une tâche d'opérations par lot S3.

Utilisation d'opérations pat lot S3 pour chiffrer des objets avec des clés de compartiment S3

Dans cette section, vous allez utiliser l'opération de copie des opérations par lot Amazon S3 pour identifier et activer le chiffrement de clés de compartiment S3 sur des objets existants. Pour de plus amples informations sur les clés de compartiment S3, veuillez consulter Réduction du coût du SSE-KMS avec les clés de compartiment Amazon S3 et Configuration de votre compartiment de sorte qu'il utilise une clé de compartiment S3 avec SSE-KMS pour de nouveaux objets.

Les rubriques abordées dans cet exemple sont les suivantes :

Rubriques

Prérequis

Pour suivre les étapes de cette procédure, vous devez avoir un Compte AWS et au moins un compartiment S3 pour accueillir vos fichiers de travail et vos résultats chiffrés. Il se peut également que vous trouviez des informations utiles dans une grande partie de la documentation existante sur les opérations par lot S3, dont les rubriques suivantes :

- Principes de base des opérations par lot S3
- Création d'une tâche d'opérations par lot S3
- Opérations prises en charge par les opérations par lot S3
- Gestion des tâches d'opérations par lot S3

Étape 1 : obtenir votre liste d'objets à l'aide d'Amazon S3 Inventory

Pour commencer, identifiez le compartiment S3 contenant les objets à chiffrer, et obtenez la liste de son contenu. Un rapport d'inventaire Amazon S3 est le moyen le plus pratique et le plus abordable de le faire. Le rapport fournit la liste des objets dans un compartiment ainsi que les métadonnées associées. Le compartiment source fait référence au compartiment inventorié, et le compartiment de destination au compartiment dans lequel vous stockez le fichier de rapport d'inventaire. Pour de plus amples informations sur les compartiments sources et de destination de l'inventaire Amazon S3, veuillez consulter Inventaire Simple Storage Service (Amazon S3).

La manière la plus simple de configurer un inventaire consiste à utiliser la AWS Management Console. Vous pouvez également utiliser l'API REST, la AWS Command Line Interface (AWS CLI) ou les kits SDK AWS. Avant de suivre ces étapes, veillez à vous connecter à la console et à ouvrir la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/. Si vous rencontrez des erreurs de refus d'autorisation, ajoutez une stratégie de compartiment à votre compartiment de destination. Pour plus d’informations, consultez Accorder des autorisations pour l'inventaire S3 et les analyses S3..

Pour obtenir une liste d'objets à l'aide d'un inventaire S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Buckets (Compartiments) et sélectionnez un compartiment contenant des objets à chiffrer.
3. Sous l'onglet Management (Gestion), accédez à la section Inventory configurations (Configurations d'inventaire), puis choisissez Create inventory configuration (Créer une configuration d'inventaire).
4. Donnez un nom à votre nouvel inventaire, entrez le nom du compartiment S3 de destination, et créez éventuellement un préfixe de destination pour Amazon S3 afin d'affecter des objets dans ce compartiment.
5. Pour Output format (Format de sortie), choisissez CSV.
6. (Facultatif) Dans la section Champs supplémentaires - facultatif, choisissez Chiffrement et les autres champs de rapport qui vous intéressent. Définissez la fréquence de livraison des rapports sur Daily (Quotidienne) afin que le premier rapport soit livré à votre compartiment plus tôt.
7. Choisissez Create (Créer) pour enregistrer votre configuration.

Amazon S3 peut prendre jusqu'à 48 heures pour livrer le premier rapport. Guettez donc l'arrivée de votre premier rapport. Après avoir reçu votre premier rapport, passez à la section suivante pour filtrer le contenu de votre rapport S3 Inventory. Si vous ne souhaitez plus recevoir de rapports d'inventaire pour ce compartiment, supprimez votre configuration d'inventaire S3. Autrement, S3 livre les rapports à une fréquence quotidienne ou hebdomadaire.

Une liste d'inventaire n'est pas une point-in-time vue unique de tous les objets. Une liste d'inventaire est un instantané évolutif des éléments d'un compartiment, qui sont finalement cohérents (par exemple, il se peut que la liste n'inclue pas certains objets récemment ajoutés ou supprimés). La combinaison de l'inventaire S3 et des opérations par lot S3 fonctionne de façon optimale lorsque vous travaillez avec des objets statiques, ou avec un ensemble d'objets que vous avez créé au moins deux jours auparavant. Pour utiliser des données plus récentes, utilisez l'opération d'API ListObjectsV2 (GET Bucket) pour créer votre liste d'objets manuellement. Si nécessaire, répétez le processus pendant quelques jours ou jusqu'à ce que votre rapport d'inventaire affiche l'état souhaité pour toutes les clés.

Étape 2 : Filtrer votre liste d'objets avec S3 Select

Après avoir reçu votre rapport S3 Inventory, vous pouvez filtrer son contenu pour répertorier uniquement les objets qui ne sont pas chiffrés avec des clés de compartiment S3. Si vous voulez que

tous les objets de votre compartiment soient chiffrés avec des clés de compartiment S3, vous pouvez ignorer cette étape. Toutefois, le filtrage de votre rapport d'inventaire S3 à ce stade vous permet d'économiser du temps et de l'argent en lien avec le re-chiffrement d'objets que vous avez chiffrés précédemment.

Bien que les étapes suivantes montrent comment filtrer à l'aide d'Amazon S3 Select, vous pouvez également utiliser Amazon Athena. Pour décider de l'outil à utiliser, consultez le fichier `manifest.json` de votre rapport d'inventaire S3. Ce fichier répertorie le nombre de fichiers de données associés à ce rapport. Si ce nombre est conséquent, utilisez le service Amazon Athena, car il s'exécute sur plusieurs objets S3, tandis que S3 Select opère sur un objet à la fois. Pour de plus amples informations sur l'utilisation d'Amazon S3 et d'Athena ensemble, veuillez consulter Interrogation d'un inventaire Amazon S3 avec Amazon Athena et Utilisation d'Athena dans le billet de blog Chiffrement d'objets avec des opérations par lot Amazon S3.

Pour filtrer votre rapport d'inventaire S3 en utilisant S3 Select

1. Ouvrez le fichier `manifest.json` à partir de votre rapport d'inventaire et consultez la section `fileSchema` du fichier JSON. Celle-ci informe la requête que vous exécutez sur les données.

   Le JSON suivant est un exemple de fichier `manifest.json` pour un inventaire au format CSV dans un compartiment pour lequel la gestion des versions est activée. L'aspect de votre manifeste peut varier selon la façon dont vous avez configuré votre rapport d'inventaire.

```
  {
    "sourceBucket": "batchoperationsdemo",
    "destinationBucket": "arn:aws:s3:::testbucket",
    "version": "2021-05-22",
    "creationTimestamp": "1558656000000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
 BucketKeyStatus",
    "files": [
      {
        "key": "demoinv/batchoperationsdemo/DemoInventory/data/009a40e4-
f053-4c16-8c75-6100f8892202.csv.gz",
        "size": 72691,
        "MD5checksum": "c24c831717a099f0ebe4a9d1c5d3935c"
      }
    ]
  }
```

Si la gestion des versions n'est pas activée sur le compartiment, ou si vous choisissez d'exécuter le rapport pour les dernières versions, le `fileSchema` est `Bucket`, `Key` et `BucketKeyStatus`.

Si la gestion des versions est activée, selon la façon dont vous avez configuré le rapport d'inventaire, le `fileSchema` peut inclure les éléments suivants : `Bucket`, `Key`, `VersionId`, `IsLatest`, `IsDeleteMarker` et `BucketKeyStatus`. Soyez donc attentif aux colonnes 1, 2, 3 et 6 lorsque vous exécutez votre requête.

Les opérations par lot S3 ont besoin du compartiment, de la clé et de l'ID de version entrés pour effectuer la tâche, en plus du champ sur lequel effectuer la recherche, qui est `BucketKeyStatus`. Vous n'avez pas besoin du champ d'ID de version, mais il est utile de le spécifier lorsque vous opérez sur un compartiment avec la gestion des versions. Pour plus d'informations, consultez Utiliser des objets dans un compartiment activé pour la gestion des versions.

2. Recherchez les fichiers de données pour le rapport d'inventaire. L'objet `manifest.json` répertorie les fichiers de données sous files.
3. Après avoir localisé et sélectionné le fichier de données dans la console S3, choisissez Actions, puis Query with S3 Select (Requête avec S3 Select).
4. Conservez la sélection des champs prédéfinis CSV,Comma, et GZIP, puis choisissez Next (Suivant).
5. Pour réviser le format de votre rapport d'inventaire avant de poursuivre, choisissez Afficher l'aperçu du fichier.
6. Saisissez les colonnes à référencer dans la zone d'expression SQL, puis sélectionnez Run SQL (Exécuter SQL). L'expression suivante renvoie les colonnes 1 à 3 pour tous les objets sans clé de compartiment S3 configurée.

```
select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
```

Voici quelques exemples de résultats.

```
batchoperationsdemo,0100059%7Ethumb.jpg,lsrtIxksLu0R0ZkYPL.LhgD5caTYn6vu
batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR
batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLnl1mXD5c4BwiOIinqFrktddkoL
batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDFlE.l3Y0
batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_sK89trmXYIwooABSh
batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor
```

```
      batchoperationsdemo,200907200065HQ
%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4
      batchoperationsdemo,200907200076HQ
%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
      batchoperationsdemo,201103190002HQ
%7Ethumb.jpg,z.2sVRh0myqVi0BuIrngWlsRPQdb7qOS
```

7. Téléchargez les résultats, enregistrez-les au format CSV, et chargez-les sur Amazon S3 en tant que liste d'objets pour la tâche d'opérations par lot S3.
8. Si vous avez plusieurs fichiers manifeste, exécutez une Requête avec S3 Select sur ceux-ci. En fonction de la taille des résultats, vous pouvez combiner les listes et exécuter un seule tâche d'opérations par lot S3, ou exécuter chaque liste en tant que tâche distincte.

   Pour décidez du nombre de tâches à exécuter, considérez le prix d'exécution de chaque tâche d'opérations par lot S3 .

Étape 3 : Configurer et exécuter votre tâche d'opérations par lot S3

À présent que vos listes CSV d'objets S3 sont filtrées, vous pouvez commencer la tâche d'opérations par lot S3 pour chiffrer les objets avec des clés de compartiment S3.

Une tâche fait référence collectivement à la liste (manifeste) d'objets fournis, à l'opération effectuée et aux paramètres spécifiés. La manière la plus simple de chiffrer cet ensemble d'objets consiste à utiliser l'opération de copie PUT en spécifiant le même préfixe de destination que celui des objets répertoriés dans le manifeste. Cela a pour effet, soit de remplacer les objets existants dans un compartiment sans gestion des versions, soit, lorsque la gestion des versions est activée, de créer une version chiffrée plus récente des objets.

Dans le cadre de la copie des objets, spécifiez qu'Amazon S3 doit chiffrer l'objet avec le chiffrement SSE-KMS et S3. Cette tâche copiant les objets, à la fin, tous vos objets présentent une date de création mise à jour, quelle que soit la date à laquelle vous les avez initialement ajoutés à S3. Dans le cadre de la tâche d'opérations par lot S3, spécifiez également les autres propriétés de votre ensemble d'objets, y compris les étiquettes d'objet et la classe de stockage.

Sous-étapes

- Configurer votre stratégie IAM
- Configurer votre rôle IAM d'opérations par lot
- Activer les clés de compartiment S3 pour un compartiment existant

- Créer votre tâche d'opérations par lot
- Exécuter votre tâche d'opérations par lot
- À savoir

Configurer votre stratégie IAM

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation, choisissez Policy (Sratégies), puis Create Policy (Créer une stratégie).
3. Choisissez l'onglet JSON. Choisissez Edit policy (Modifier une stratégie), puis ajoutez l'exemple de stratégie IAM qui apparaît dans le bloc de code suivant.

   Après avoir copié l'exemple de politique dans votre console IAM, remplacez ce qui suit :

   a. Remplacez *SOURCE_BUCKET_FOR_COPY* par le nom de votre compartiment source.
   b. Remplacez *DESTINATION_BUCKET_FOR_COPY* par le nom de votre compartiment de destination.
   c. Remplacez *MANIFEST_KEY* par le nom de votre objet manifeste.
   d. Remplacez *REPORT_BUCKET* par le nom du compartiment dans lequel vous voulez enregistrer les rapports.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CopyObjectsToEncrypt",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectTagging",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectVersionAcl",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
```

```
          "s3:GetObjectVersionTagging"
        ],
        "Resource": [
          "arn:aws:s3:::SOURCE_BUCKET_FOR_COPY/*",
          "arn:aws:s3:::DESTINATION_BUCKET_FOR_COPY/*"
        ]
      },
      {
        "Sid": "ReadManifest",
        "Effect": "Allow",
        "Action": [
          "s3:GetObject",
          "s3:GetObjectVersion"
        ],
        "Resource": "arn:aws:s3:::MANIFEST_KEY"
      },
      {
        "Sid": "WriteReport",
        "Effect": "Allow",
        "Action": [
          "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::REPORT_BUCKET/*"
      }
    ]
  }
```

4. Choisissez Next: Tags (Suivant : Étiquettes).
5. Ajoutez les étiquettes de votre choix (facultatif), puis choisissez Next: Review (Suivant : Vérification).
6. Ajoutez un nom de stratégie et, éventuellement, une description, puis choisissez Create policy (Créer une stratégie).
7. Choisissez Review policy (Examiner une stratégie), puis Save changes (Enregistrer les modifications).
8. Une fois votre stratégie d'opérations par lot S3 prête, la console vous renvoie à l'IAM Policies (Stratégies). Filtrez le nom de stratégie, choisissez le bouton situé à gauche du nom de stratégie, choisissez Policy actions (Actions de stratégie), puis Attach (Attacher).

   Pour attacher la stratégie nouvellement créée à un rôle IAM, sélectionnez les utilisateurs, groupes ou rôles appropriés dans votre compte, puis choisissez Attach Policy (Attacher une stratégie). Cela a pour effet de vous ramener à la console IAM.

Configurer votre rôle IAM d'opérations par lot

1. Sur la console IAM, dans le volet de navigation, sélectionnez Rôles, puis sélectionnez Créer un rôle.
2. Sélectionnez Service AWS, S3 et Opérations par lots S3. Choisissez ensuite Next: Permissions (Suivant : Autorisations).
3. Commencer à entrer le nom de la stratégie IAM que vous venez de créer. Activez la case à cocher en regard du nom de stratégie quand il s'affiche, puis choisissez Next: Tags (Suivant : Étiquettes).
4. (Facultatif) Ajoutez des étiquettes ou gardez les champs de clé et de valeur vides pour cet exercice. Choisissez Next: Review (Suivant : Vérification).
5. Entrez un nom de rôle, puis acceptez la description par défaut ou ajoutez la vôtre. Sélectionnez Create role (Créer un rôle).
6. Assurez-vous que l'utilisateur qui crée la tâche dispose des autorisations décrites dans l'exemple suivant.

   Remplacez *{ACCOUNT-ID}* par l'ID de votre Compte AWS, et *{IAM_ROLE_NAME}* par le nom que vous envisagez d'appliquer au rôle IAM que vous allez créer ultérieurement à l'étape de création de tâche d'opérations par lot. Pour plus d'informations, consultez Octroi d'autorisations pour les opérations par lot Simple Storage Service (Amazon S3).

   ```
   {
   "Sid": "AddIamPermissions",
   "Effect": "Allow",
   "Action": [
   "iam:GetRole",
   "iam:PassRole"
   ],
   "Resource": "arn:aws:iam::ACCOUNT-ID:role/IAM_ROLE_NAME"
   }
   ```

Activer les clés de compartiment S3 pour un compartiment existant

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le compartiment pour lequel vous souhaitez activer une clé de compartiment S3.
3. Choisissez Propriétés.

4. Sous Default encryption (Chiffrement par défaut), choisissez Edit (Modifier).
5. Sous Type de chiffrement, vous avez le choix entre Clés gérées par Amazon S3 (SSE-S3) et Clé AWS Key Management Service (SSE-KMS).
6. Si vous avez choisi Clé AWS Key Management Service (SSE-KMS), sous AWS KMS key, vous pouvez spécifier la clé AWS KMS via l'une des options suivantes.

   - Pour choisir parmi la liste des clés KMS disponibles, sélectionnez Choisir parmi vos clés AWS KMS. Dans la liste des clés disponibles, choisissez une clé KMS symétrique de chiffrement dans la même région que votre compartiment. La clé gérée par AWS (`aws/s3`) et votre clé gérée par le client apparaissent toutes deux dans la liste.
   - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de clé AWS KMS, puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
   - Pour créer une clé gérée par le client dans la console AWS KMS, choisissez Créer une clé KMS.
7. Sous Bucket Key (Clé de compartiment), choisissez Enable (Activer), puis Save changes (Enregistrer les modifications).

À présent que la clé de compartiment S3 est activée au niveau du compartiment, les objets chargés, modifiés ou copiés dans ce compartiment hériteront de cette configuration de chiffrement par défaut. Cela inclut les objets copiés à l'aide d'opérations par lot Amazon S3.

Créer votre tâche d'opérations par lot

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, choisissez Opérations par lot, puis Créer une tâche.
3. Choisissez la Région dans laquelle vous stockez vos objets, puis choisissez CSV comme type de manifeste.
4. Saisissez le chemin d'accès ou accédez au fichier manifeste CSV que vous avez créé précédemment à partir des résultats de S3 Select (ou d'Athena). Si votre manifeste contient des ID de version, cochez cette case. Choisissez Suivant.
5. Choisissez l'opération Copy (Copier), le compartiment de destination de la copie. Vous pouvez maintenir le chiffrement côté serveur désactivé. Tant que les clés de compartiment S3 sont activées pour la destination du compartiment, l'opération de copie applique ces clés à ce compartiment.

6. (Facultatif) Choisissez une classe de stockage et les autres paramètres souhaités. Les paramètres que vous spécifiez dans cette étape s'appliquent à toutes les opérations effectuées sur les objets répertoriés dans le manifeste. Choisissez Suivant.
7. Pour configurer le chiffrement côté serveur, procédez comme suit :

   a. Sous Chiffrement côté serveur, choisissez l'une des options suivantes :

      - Pour conserver les paramètres du compartiment pour le chiffrement côté serveur par défaut des objets lors de leur stockage dans Amazon S3, choisissez Ne pas spécifier de clé de chiffrement. Tant que les clés de compartiment S3 sont activées pour la destination du compartiment, l'opération de copie applique une clé de compartiment S3 au compartiment de destination.

        > **Note**
        >
        > Si la politique de compartiment pour la destination spécifiée exige que les objets soient chiffrés avant de les stocker dans Amazon S3, vous devez spécifier une clé de chiffrement. Sinon, la copie des objets vers la destination échouera.

      - Pour chiffrer des objets avant de les stocker dans Amazon S3, choisissez Spécifier une clé de chiffrement.

   b. Dans Paramètres de chiffrement, si vous choisissez Spécifier une clé de chiffrement, vous devez choisir Utiliser les paramètres du compartiment de destination pour le chiffrement par défaut ou Ignorer les paramètres du compartiment de destination pour le chiffrement par défaut.

   c. Si vous choisissez Ignorer les paramètres du compartiment de destination pour le chiffrement par défaut, vous devez configurer les paramètres de chiffrement suivants.

      i. Sous Type de chiffrement, vous devez choisir Clés gérées par Amazon S3 (SSE-S3) ou Clé AWS Key Management Service (SSE-KMS). SSE-S3 utilise l'un des chiffrements par bloc les plus puissants qui existent, Advanced Encryption Standard à 256 bits (AES-256) pour chiffrer chaque objet. SSE-KMS vous permet de mieux contrôler votre clé. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) et Utilisation du chiffrement côté serveur à l'aide de AWS KMS clés (SSE-KMS).

      ii. Si vous choisissez Clé AWS Key Management Service (SSE-KMS), sous AWS KMS key, vous pouvez spécifier votre AWS KMS key via l'une des options suivantes.

- Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos AWS KMS keys, puis sélectionnez une clé KMS de chiffrement symétrique dans la même région que votre compartiment. La clé gérée par AWS (`aws/s3`) et votre clé gérée par le client apparaissent toutes deux dans la liste.
- Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de clé AWS KMS, puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
- Pour créer une clé gérée par le client dans la console AWS KMS, choisissez Créer une clé KMS.

iii. Sous Clé de compartiment, choisissez Activer. L'opération de copie applique une clé de compartiment S3 au compartiment de destination.

8. Saisissez une description pour votre tâche (ou conservez la description par défaut), définissez son niveau de priorité, choisissez un type de rapport, puis spécifiez le Path to completion report destination (Chemin d'accès de la destination du rapport de fin de tâche).
9. Dans la section Permissions (Autorisations), veillez à choisir le rôle IAM d'opérations par lot que vous avez défini précédemment. Choisissez Suivant.
10. Sous Review (Vérification), vérifiez les paramètres. Si vous voulez apporter des modifications, choisissez Previous (Précédent). Après avoir confirmé les paramètres d'opérations par lot, choisissez Create job (Créer une tâche).

    Pour de plus amples informations, veuillez consulter Création d'une tâche d'opérations par lot S3.

Exécuter votre tâche d'opérations par lot

L'assistant de configuration vous renvoie automatiquement à la section Opérations par lot S3 de la console Amazon S3. Votre nouvelle tâche passe de l'état New (Nouvelle) à l'état Preparing (Préparation) quand S3 commence le processus. Pendant la phase de préparation, S3 lit le manifeste de la tâche, vérifie s'il contient des erreurs, et calcule le nombre d'objets.

1. Choisissez le bouton d'actualisation dans la console Amazon S3 pour vérifier la progression. Selon la taille du manifeste, la lecture peut prendre des minutes ou des heures.
2. Quand S3 finit de lire le manifeste de la tâche, la tâche passe à l'état Awaiting your confirmation (En attente de confirmation). Cliquez sur le bouton d'option à gauche de l'ID de tâche, puis choisissez Run job (Exécuter la tâche).

3. Vérifiez les paramètres de la tâche, puis, dans le coin inférieur droit choisissez Run job (Exécuter la tâche).

   Lorsque la tâche commence à s'exécuter, vous pouvez choisir le bouton d'actualisation pour vérifier sa progression dans l'affichage du tableau de bord de la console ou en sélectionnant la tâche.
4. Une fois la tâche est terminée, vous pouvez afficher les informationsSuccèsandÉcheccompte pour confirmer que tout a été effectué comme prévu. Si vous avez activé les rapports de tâche, vérifiez dans votre rapport de tâche la cause exacte de toute opération ayant échoué.

   Vous pouvez également effectuer ces étapes en utilisant l'AWS CLI, les kits SDK AWS ou l'API REST Amazon S3. Pour plus d'informations sur le suivi de l'état des tâches et les rapports de fin de tâche, veuillez consulterSuivi de l'état de la tâche et des rapports de fin de tâche.

À savoir

Lorsque vous utilisez les opérations par lot S3 pour chiffrer des objets avec des clés de compartiment S3, tenez compte des points suivants :

- Vous serez facturé pour les tâches, objets et demandes d'opérations par lot S3 en plus des frais associés à ce que les opérations par lot S3 accomplissent en votre nom, dont les transferts de données, les demandes, ainsi que d'autres frais. Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
- Si vous utilisez un compartiment avec gestion des versions, chaque tâche d'opérations par lot S3 effectuée crée de nouvelles versions chiffrées de vos objets. Elle conserve également les versions précédentes sans clé de compartiment S3 configurée. Pour supprimer les anciennes versions, configurez une stratégie d'expiration du cycle de vie S3 pour les versions non actuelles, comme décrit dans Éléments de la configuration du cycle de vie.
- L'opération de copie crée de nouveaux objets avec de nouvelles dates de création, ce qui peut affecter des actions du cycle de vie telles que l'archivage. Si vous copiez tous les objets dans votre compartiment, toutes les nouvelles copies ont des dates de création identiques ou similaires. Pour mieux identifier ces objets et créer différentes règles de cycle de vie pour différents sous-ensembles de données, envisagez d'utiliser des étiquettes d'objet.

Récapitulatif

Dans cette section, vous avez trié les objets existants pour filtrer les données déjà chiffrées. Ensuite, vous avez appliqué la fonction de clé de compartiment S3 sur des objets non chiffrés en utilisant

des opérations par lot S3 pour copier les données existantes vers un compartiment avec la clé de compartiment S3 activée. Ce processus peut vous faire gagner du temps et de l'argent tout en vous permettant d'effectuer des opérations telles que le chiffrement de tous les objets existants.

Pour de plus amples informations sur les opérations par lot S3, veuillez consulter Exécution des opérations par lot à grande échelle sur des objets Amazon S3.

Pour prendre connaissance d'exemples montrant l'opération de copie avec des balises à l'aide de la AWS CLI et du kit AWS SDK for Java, consultez Création d'une tâche d'opérations par lot avec des étiquettes de tâche utilisées pour l'étiquetage.

AWS Lambda Fonction Invoke

La AWS Lambda fonction Invoke lance des AWS Lambda fonctions pour effectuer des actions personnalisées sur les objets répertoriés dans un manifeste. Cette section décrit comment créer une fonction Lambda à utiliser avec des opérations par lot S3 et comment créer une tâche pour appeler la fonction. La tâche S3 Batch Operations utilise l'opération `LambdaInvoke` pour exécuter une fonction Lambda sur chaque objet répertorié dans un manifeste.

Vous pouvez utiliser S3 Batch Operations for Lambda à l'aide des AWS Management Console API, AWS Command Line Interface (AWS CLI), AWS SDK ou REST. Pour plus d'informations sur l'utilisation de Lambda, consultez la section Mise en route avec AWS Lambda du Guide du développeur AWS Lambda .

Les sections suivantes expliquent comment commencer à utiliser les opérations par lot S3 avec Lambda.

Rubriques

- Utilisation de Lambda avec les opérations par lot Amazon S3
- Création d'une fonction Lambda à utiliser avec les opérations par lot S3
- Création d'une tâche d'opérations par lot S3 qui appelle une fonction Lambda
- Fourniture d'informations au niveau des tâches dans les manifestes Lambda
- Apprendre du tutoriel sur les opérations par lots S3

Utilisation de Lambda avec les opérations par lot Amazon S3

Lorsque vous utilisez S3 Batch Operations avec AWS Lambda, vous devez créer de nouvelles fonctions Lambda spécifiquement destinées à être utilisées avec S3 Batch Operations. Vous ne pouvez pas réutiliser les fonctions basées sur des événements Amazon S3 existantes avec les

opérations par lot S3. Les fonctions d'événements peuvent seulement recevoir des messages. Elles ne peuvent pas en renvoyer. Les fonctions Lambda utilisées avec les opérations par lot S3 doivent accepter et renvoyer des messages. Pour plus d'informations sur l'utilisation de Lambda avec les événements Amazon S3, consultez la section Utilisation AWS Lambda avec Amazon S3 dans le manuel du AWS Lambda développeur.

Vous créez une tâche d'opérations par lot S3 qui appelle votre fonction Lambda. La tâche exécute la même fonction Lambda pour tous les objets répertoriés dans votre manifeste. Vous pouvez contrôler les versions de votre fonction Lambda à utiliser lors du traitement des objets dans votre manifeste. Les opérations par lot S3 prennent en charge les Amazon Resource Names (ARN) non qualifiés, les alias et les versions spécifiques. Pour plus d'informations, consultez la section Présentation de la AWS Lambda gestion des versions du Guide du développeur AWS Lambda .

Si vous donnez un ARN de fonction qui utilise un alias ou le qualificatif `$LATEST` à la tâche d'opérations par lot S3, et que vous mettez à jour la version vers laquelle l'un de ces points pointe, les opérations par lot S3 commencent à appeler la nouvelle version de votre fonction Lambda. Cela peut être utile lorsque vous souhaitez mettre à jour la fonctionnalité en cours de traitement d'une tâche importante. Si vous ne souhaitez pas que les opérations par lot S3 modifient la version utilisée, indiquez la version spécifique dans le paramètre `FunctionARN` lorsque vous créez votre tâche.

Utilisation de Lambda et des opérations par lots Amazon S3 avec des compartiments de répertoires

Les compartiments de répertoires sont un type de compartiment Amazon S3 conçu pour les charges de travail ou les applications critiques en termes de performances qui nécessitent une latence constante inférieure à dix millisecondes. Pour plus d'informations, consultez Compartiments de répertoires.

L'utilisation d'opérations par lots Amazon S3 pour invoquer des fonctions Lambda agissant sur des compartiments de répertoires est soumise à des exigences particulières. Par exemple, vous devez structurer votre demande Lambda à l'aide d'un schéma JSON mis à jour et spécifier `InvocationSchemaVersion` **`2.0`** lorsque vous créez la tâche. Ce schéma mis à jour vous permet de spécifier des paires clé-valeur facultatives pour `UserArguments`, que vous pouvez utiliser pour modifier certains paramètres des fonctions Lambda existantes. Pour plus d'informations, consultez Automatiser le traitement des objets dans les compartiments d'annuaire Amazon S3 avec S3 Batch Operations et AWS Lambda sur le blog AWS de stockage.

Codes de réponse et de résultat

S3 Batch Operations invoque la fonction Lambda avec une ou plusieurs clés, chacune étant associée à `TaskID` une clé. S3 Batch Operations attend un code de résultat par clé de la part des fonctions

Lambda. Tous les identifiants de tâche envoyés dans la demande qui ne sont pas renvoyés avec un code de résultat par clé recevront le code de résultat indiqué `treatMissingKeysAs` dans le champ. `treatMissingKeysAs`est un champ de demande facultatif dont la valeur par défaut est. `TemporaryFailure` Le tableau suivant contient les autres codes de résultat et valeurs possibles pour le `treatMissingKeysAs` champ.

Code de réponse	Description
`Succeeded`	La tâche s'est achevée normalement. Si vous avez demandé un rapport de fin de tâche, la chaîne de résultat de la tâche est comprise dans le rapport.
`TemporaryFailure`	La tâche a fait l'objet d'un échec temporair e et sera relancée avant la fin de la tâche. La chaîne de résultat est ignorée. Si c'est le relancement final, le message d'erreur est inclus dans le rapport final.
`PermanentFailure`	La tâche a fait l'objet d'un échec permanent. Si vous avez demandé un rapport de fin de tâche, la tâche est marquée comme `Failed` et inclut la chaîne de message d'erreur. Les chaînes de résultats provenant de tâches échouées sont ignorées.

Création d'une fonction Lambda à utiliser avec les opérations par lot S3

Cette section fournit des exemples d'autorisations AWS Identity and Access Management (IAM) que vous devez utiliser avec votre fonction Lambda. Elle contient également un exemple de fonction Lambda à utiliser avec les opérations par lot S3. Si vous n'avez jamais créé de fonction Lambda auparavant, consultez Tutoriel : Utilisation AWS Lambda avec Amazon S3 dans le Guide du AWS Lambda développeur.

Vous devez créer des fonctions Lambda spécifiques à utiliser avec les opérations par lot S3. Vous ne pouvez pas réutiliser de fonctions Lambda existantes basées sur des événements Amazon S3. En

effet, les fonctions Lambda utilisées pour les opérations par lot S3 doivent accepter et renvoyer des champs de données spéciaux.

> **Important**
>
> AWS Lambda les fonctions écrites en Java acceptent l'une RequestHandlerou l'autre des RequestStreamHandlerinterfaces de gestion. Cependant, pour prendre en charge le format de demande et de réponse S3 Batch Operations, AWS Lambda il faut une `RequestStreamHandler` interface pour la sérialisation personnalisée et la désérialisation d'une demande et d'une réponse. Cette interface permet à Lambda de transmettre un « InputStream and » OutputStream à la méthode Java`handleRequest`.
> Assurez-vous d'utiliser l'interface `RequestStreamHandler` lorsque vous utilisez les fonctions Lambda avec les opérations par lot S3. Si vous utilisez une interface `RequestHandler`, la tâche par lot échouera avec « JSON non valide retourné dans la charge utile Lambda » dans le rapport de fin.
> Pour plus d'informations, consultez Interfaces du gestionnaire dans le Guide de l'utilisateur AWS Lambda .

Exemples d'autorisations IAM

Voici des exemples d'autorisations IAM nécessaires à l'utilisation d'une fonction Lambda avec les opérations par lot S3.

Example - Stratégie d'approbation des opérations par lot S3.

Voici un exemple de stratégie d'approbation que vous pouvez utiliser pour le rôle IAM des opérations par lot. Ce rôle IAM est spécifié lorsque vous créez la tâche et donne aux opérations par lot l'autorisation d'endosser le rôle IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "batchoperations.s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
```

```
    ]
}
```

Example - Stratégie d'IAM Lambda

Voici un exemple de stratégie IAM qui donne aux opérations par lot S3 l'autorisation d'appeler la fonction Lambda et de lire le manifeste d'entrée.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BatchOperationsLambdaPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:PutObject",
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemple de demande et de réponse

Cette section fournit des exemples de demandes et réponses pour la fonction Lambda.

Example Demande

L'exemple suivant est un exemple JSON de requête pour la fonction Lambda.

```
{
    "invocationSchemaVersion": "1.0",
    "invocationId": "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
    "job": {
        "id": "f3cc4f60-61f6-4a2b-8a21-d07600c373ce"
    },
    "tasks": [
        {
            "taskId": "dGFza2lkZ29lc2hlcmUK",
            "s3Key": "customerImage1.jpg",
```

```
            "s3VersionId": "1",
            "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:awsexamplebucket1"
        }
    ]
}
```

Example Réponse

L'exemple suivant est un exemple JSON de réponse pour la fonction Lambda.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
      "resultString": "[\"Mary Major", \"John Stiles\"]"
    }
  ]
}
```

Exemple de fonction Lambda pour les opérations par lot S3

L'exemple suivant Python Lambda supprime un marqueur de suppression d'un objet versionné.

Comme l'exemple le montre, les clés des opérations par lot S3 sont encodées par URL. Pour utiliser Amazon S3 avec d'autres AWS services, il est important de décoder par URL la clé transmise par S3 Batch Operations.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")


def lambda_handler(event, context):
```

```
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]

    try:
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
obj_key
        )

        try:
            # If this call does not raise an error, the object version is not a delete
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "PermanentFailure"
            result_string = (
                f"Object {obj_key}, ID {obj_version_id} is not " f"a delete marker."
            )

            logger.debug(response)
            logger.warning(result_string)
```

```
        except ClientError as error:
            delete_marker = error.response["ResponseMetadata"]["HTTPHeaders"].get(
                "x-amz-delete-marker", "false"
            )
            if delete_marker == "true":
                logger.info(
                    "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
                )
                try:
                    s3.delete_object(
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                    )
                    result_code = "Succeeded"
                    result_string = (
                        f"Successfully removed delete marker "
                        f"{obj_version_id} from object {obj_key}."
                    )
                    logger.info(result_string)
                except ClientError as error:
                    # Mark request timeout as a temporary failure so it will be
retried.
                    if error.response["Error"]["Code"] == "RequestTimeout":
                        result_code = "TemporaryFailure"
                        result_string = (
                            f"Attempt to remove delete marker from  "
                            f"object {obj_key} timed out."
                        )
                        logger.info(result_string)
                    else:
                        raise
            else:
                raise ValueError(
                    f"The x-amz-delete-marker header is either not "
                    f"present or is not 'true'."
                )
    except Exception as error:
        # Mark all other exceptions as permanent failures.
        result_code = "PermanentFailure"
        result_string = str(error)
        logger.exception(error)
    finally:
        results.append(
            {
```

```
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Création d'une tâche d'opérations par lot S3 qui appelle une fonction Lambda

Lors de la création d'une tâche d'opérations par lot S3 pour appeler une fonction Lambda, vous devez fournir les éléments suivants :

- L'ARN de votre fonction Lambda (qui peut inclure l'alias de fonction ou le numéro de version spécifique)
- Un rôle IAM doté de l'autorisation d'invoquer la fonction
- Le de paramètre d'actio `LambdaInvokeFunction`

Pour plus d'informations sur la création d'une tâche d'opérations par lots S3, consultez Création d'une tâche d'opérations par lot S3 et Opérations prises en charge par les opérations par lot S3.

L'exemple suivant crée une tâche d'opérations par lot S3 qui appelle une fonction Lambda à l'aide de la AWS CLI.

```
aws s3control create-job
    --account-id <AccountID>
    --operation  '{"LambdaInvoke": { "FunctionArn":
 "arn:aws:lambda:Region:AccountID:function:LambdaFunctionName" } }'
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":
{"ObjectArn":"arn:aws:s3:::ManifestLocation","ETag":"ManifestETag"}}'
    --report
 '{"Bucket":"arn:aws:s3:::awsexamplebucket1","Format":"Report_CSV_20180820","Enabled":true,"Pre
```

```
    --priority 2
    --role-arn arn:aws:iam::AccountID:role/BatchOperationsRole
    --region Region
    --description “Lambda Function"
```

Fourniture d'informations au niveau des tâches dans les manifestes Lambda

Lorsque vous utilisez AWS Lambda des fonctions avec S3 Batch Operations, vous souhaiterez peut-être que des données supplémentaires accompagnent chaque tâche/clé exécutée. Par exemple, vous pouvez souhaiter que la clé d'objet source et la nouvelle clé d'objet soient fournies. Votre fonction Lambda peut alors copier la clé source vers un nouveau compartiment S3 avec un nouveau nom. Par défaut, les opérations par lot Amazon S3 vous permettent de spécifier uniquement le compartiment de destination et une liste de clés source dans le manifeste d'entrée dans votre tâche. La section suivante décrit comment inclure des données supplémentaires dans votre manifeste afin que vous puissiez exécuter des fonctions Lambda plus complexes.

Pour spécifier des paramètres individuels pour chaque clé dans votre manifeste d'opérations par lot S3 à utiliser dans le code de votre fonction Lambda, utilisez le format JSON codé en URL suivant. Le champ `key` est transmis à votre fonction Lambda comme s'il s'agissait d'une clé d'objet Amazon S3. La fonction Lambda peut cependant considérer qu'il contient d'autres valeurs ou plusieurs clés, comme illustré ci-dessous.

> **Note**
>
> Dans le manifeste, le nombre maximal de caractères pour le champ `key` est 1 024.

Example - manifeste substituant les « clés Amazon S3 » par des chaînes JSON

La version codée URL est fournie aux opérations par lot S3.

```
my-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
my-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
my-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example - manifeste codé en URL

Cette version codée en URL doit être fournie aux opérations par lot S3. La version non codée en URL ne fonctionne pas.

```
my-bucket,%7B%22origKey%22%3A%20%22object1key%22%2C%20%22newKey%22%3A
%20%22newObject1Key%22%7D
my-bucket,%7B%22origKey%22%3A%20%22object2key%22%2C%20%22newKey%22%3A
%20%22newObject2Key%22%7D
my-bucket,%7B%22origKey%22%3A%20%22object3key%22%2C%20%22newKey%22%3A
%20%22newObject3Key%22%7D
```

Example - Fonction Lambda avec format de manifeste écrivant les résultats dans le rapport de la tâche

Cet exemple de manifeste codé par URL contient des clés d'objet délimitées par des canaux que la fonction Lambda suivante doit analyser.

```
my-bucket,object1key%7Clower
my-bucket,object2key%7Cupper
my-bucket,object3key%7Creverse
my-bucket,object4key%7Cdelete
```

Cette fonction Lambda montre comment analyser une tâche séparée par une barre verticale qui est codé dans le manifeste des opérations par lot S3. La tâche indique l'opération de révision qui est appliquée à l'objet spécifié.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.resource("s3")


def lambda_handler(event, context):
    """
    Applies the specified revision to the specified object.

    :param event: The Amazon S3 batch event that contains the ID of the object to
                  revise and the revision type to apply.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation.
```

```
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]
    # The revision type is packed with the object key as a pipe-delimited string.
    obj_key, revision = parse.unquote(task["s3Key"], encoding="utf-8").split("|")
    bucket_name = task["s3BucketArn"].split(":")[-1]

    logger.info("Got task: apply revision %s to %s.", revision, obj_key)

    try:
        stanza_obj = s3.Bucket(bucket_name).Object(obj_key)
        stanza = stanza_obj.get()["Body"].read().decode("utf-8")
        if revision == "lower":
            stanza = stanza.lower()
        elif revision == "upper":
            stanza = stanza.upper()
        elif revision == "reverse":
            stanza = stanza[::-1]
        elif revision == "delete":
            pass
        else:
            raise TypeError(f"Can't handle revision type '{revision}'.")

        if revision == "delete":
            stanza_obj.delete()
            result_string = f"Deleted stanza {stanza_obj.key}."
        else:
            stanza_obj.put(Body=bytes(stanza, "utf-8"))
            result_string = (
                f"Applied revision type '{revision}' to " f"stanza {stanza_obj.key}."
            )

        logger.info(result_string)
        result_code = "Succeeded"
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchKey":
```

```
            result_code = "Succeeded"
            result_string = (
                f"Stanza {obj_key} not found, assuming it was deleted "
                f"in an earlier revision."
            )
            logger.info(result_string)
        else:
            result_code = "PermanentFailure"
            result_string = (
                f"Got exception when applying revision type '{revision}' "
                f"to {obj_key}: {error}."
            )
            logger.exception(result_string)
    finally:
        results.append(
            {
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Apprendre du tutoriel sur les opérations par lots S3

Le didacticiel suivant présente des end-to-end procédures complètes pour certaines tâches d'opérations par lots avec Lambda.

- Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Replace all object tags (Remplacer toutes les étiquettes des objets)

L'opération Replace all object tags (Remplacer toutes les étiquettes des objets) remplace les étiquettes Amazon S3 sur chaque objet répertorié dans le manifeste. Une étiquette d'objet Amazon S3 est une paire clé/valeur de chaînes que vous pouvez utiliser pour stocker les métadonnées relatives à un objet.

Pour créer une tâche de remplacement de toutes les étiquettes des objets, vous fournissez un ensemble de étiquettes à appliquer. S3 Batch Operations applique le même ensemble d'étiquettes à chaque objet. Le jeu d'étiquettes que vous fournissez remplace les jeux d'étiquettes déjà associés aux objets du manifeste. S3 Batch Operations ne permet pas d'ajouter des étiquettes aux objets tout en laissant les étiquettes existantes en place.

Si les objets de votre manifeste se trouvent dans un compartiment versionné, vous pouvez appliquer l'ensemble d'étiquettes à des versions spécifiques de chaque objet. Pour ce faire, spécifiez un ID de version pour chaque objet dans le manifeste. Si vous n'incluez pas d'ID de version pour les objets, Batch Operations S3 applique l'ensemble d'étiquettes à la dernière version de l'objet.

Limites et restrictions

- Le rôle IAM AWS Identity and Access Management que vous spécifiez pour exécuter la tâche Batch Operations doit disposer des autorisations nécessaires pour l'opération Amazon S3 sous-jacente de remplacement de toutes les étiquettes d'objets. Pour plus d'informations sur les autorisations requises, consultez PutObjectTagging dans la référence de l'API Amazon Simple Storage Service.
- S3 Batch Operations utilise l'opération Amazon S3 PutObjectTagging pour appliquer des étiquettes à chaque objet du manifeste. Toutes les restrictions et limitations qui s'appliquent à l'opération sous-jacente s'appliquent également aux tâches S3 Batch Opérations.

Pour plus d'informations sur l'utilisation de la console pour créer des tâches, consultez Création d'une tâche d'opérations par lot S3.

Pour plus d'informations sur le balisage des objets, consultezCatégorisation de votre stockage à l'aide de balises dans ce guide et PutObjectTagging, GetObjectTagginget DeleteObjectTagging dans la Référence de l'API Amazon Simple Storage Service.

Delete all object tags (Supprimer toutes les étiquettes des objets)

L'opération Delete all object tags (Supprimer toutes les étiquettes des objets) supprime tous les ensembles de étiquettes Amazon S3 associés aux objets répertoriés dans le manifeste. S3 Batch Operations ne permet pas de supprimer les étiquettes des objets tout en conservant les autres étiquettes en place.

Si les objets de votre manifeste se trouvent dans un compartiment versionné, vous pouvez supprimer les ensembles d'étiquettes d'une version d'un objet. Pour ce faire, spécifiez un ID de version pour chaque objet dans le manifeste. Si vous n'incluez pas d'ID de version pour un objet, S3 Batch Operations supprime l'ensemble d'étiquettes de la dernière version de chaque objet.

Pour plus d'informations sur les manifestes Batch Operations, consultez Spécification d'un manifeste.

> **Warning**
>
> L'exécution de cette tâche supprime tous les ensembles d'étiquettes sur chaque objet répertorié dans le manifeste.

Limites et restrictions

- Le rôle AWS Identity and Access Management (IAM) que vous spécifiez pour exécuter la tâche doit disposer des autorisations nécessaires pour exécuter l'opération Amazon S3 sous-jacente de suppression des étiquettes d'objets. Pour plus d'informations, consultez DeleteObjectTagging dans la référence de l'API Amazon Simple Storage Service.
- S3 Batch Operations utilise l'opération Amazon S3 DeleteObjectTagging pour supprimer les ensembles d'étiquettes de chaque objet du manifeste. Toutes les restrictions et limitations qui s'appliquent à l'opération sous-jacente s'appliquent également aux tâches S3 Batch Opérations.

Pour plus d'informations sur la création de tâches, consultez Création d'une tâche d'opérations par lot S3.

Pour plus d'informations sur le balisage des objets, reportez-vous Replace all object tags (Remplacer toutes les étiquettes des objets) dans ce guide et à PutObjectTagging, GetObjectTagginget DeleteObjectTagging dans la référence de l'API Amazon Simple Storage Service.

Replace access control list (Remplacer la liste de contrôle d'accès (ACL)

L'opération Replace access control list (ACL) (Remplacer la liste de contrôle d'accès (ACL) remplace les listes de contrôle d'accès (ACL) Amazon S3 de chaque objet répertorié dans le manifeste. Avec les ACL, vous pouvez définir les personnes qui peuvent accéder à un objet et quelles actions elles peuvent exécuter.

Les opérations par lot S3 prennent en charge les listes de contrôle d'accès personnalisées que vous spécifiez et les listes de contrôle d'accès prédéfinies fournies par Amazon S3 avec un ensemble prédéfini d'autorisations d'accès.

Si les objets de votre manifeste se trouvent dans un compartiment versionné, vous pouvez appliquer les ACL à des versions spécifiques de chaque objet. Pour ce faire, spécifiez un ID de version pour chaque objet dans le manifeste. Si vous n'incluez aucun ID de version pour les objets, les opérations par lot S3 appliquent la liste de contrôle d'accès à la dernière version de l'objet.

Pour en savoir plus sur les listes ACL dans Amazon S3, Présentation de la liste de contrôle d'accès (ACL).

S3 Block Public Access

Si vous souhaitez limiter l'accès public à tous les objets d'un compartiment, utilisez le blocage de l'accès public Amazon S3 au lieu des opérations par lot S3. Le blocage de l'accès publique permet de limiter l'accès par compartiment ou par compte via une seule opération simple et rapide. Cette option est recommandée lorsque votre objectif est de contrôler l'accès public à tous les objets d'un compartiment ou d'un compte. Utilisez S3 Batch Operations lorsque vous devez appliquer une liste de contrôle d'accès (ACL) prédéfinie à chaque objet du manifeste. Pour de plus amples informations sur le blocage de l'accès public S3, consultez Blocage de l'accès public à votre stockage Amazon S3.

Propriété de l'objet S3

Si les objets du manifeste se trouvent dans un compartiment qui utilise le paramètre appliqué par le propriétaire du compartiment pour Object Ownership, l'opération Replace access control list (ACL) (Remplacer la liste de contrôle d'accès (ACL)) ne peut spécifier que les listes ACL d'objet qui accordent le contrôle complet au propriétaire du compartiment. L'opération ne peut pas accorder d'autorisations ACL d'objet à d'autres Comptes AWS ou groupes. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Restrictions et limitations

- Le rôle que vous spécifiez pour exécuter la tâche de remplacement de liste de contrôle d'accès doit disposer des autorisations nécessaires pour exécuter l'opération Amazon S3 `PutObjectAcl` sous-jacente. Pour plus d'informations sur les autorisations requises, consultez PutObjectAcl dans la référence de l'API Amazon Simple Storage Service.
- Les opérations par lot S3 utilisent l'opération Amazon S3 `PutObjectAcl` pour appliquer la liste de contrôle d'accès ACL spécifiée à chaque objet du manifeste. Par conséquent, toutes les restrictions et limitations qui s'appliquent à l'opération PutObjectAcl sous-jacente s'appliquent également aux tâches de remplacement de liste de contrôle d'accès S3 Batch Operations.

Restauration d'objets à l'aide d'opérations par lot

L'opération Restaurer lance des demandes de restauration pour les objets Amazon S3 archivés, répertoriés dans votre manifeste. Les objets archivés suivants doivent être restaurés pour qu'il soit possible d'y accéder en temps réel :

- Objets archivés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive
- Objets archivés via la classe de stockage S3 Intelligent-Tiering dans les niveaux Archive Access ou Deep Archive Access

L'utilisation d'une opération de lancement de restauration S3 dans votre tâche S3 Batch Operations entraîne une demande de restauration pour chaque objet spécifié dans le manifeste.

> **Important**
>
> La tâche S3 Initiate Restore Object (Lancer une restauration d'objet S3) lance uniquement la demande de restauration d'objet. S3 Batch Operations indique que la tâche est terminée pour chaque objet après que la demande a été lancée pour l'objet. Amazon S3 ne met pas à jour la tâche ou ne vous avertit pas lorsque les objets ont été restaurés. Toutefois, vous pouvez utiliser des notifications d'événements S3 pour recevoir des notifications lorsque les objets sont disponibles dans Amazon S3. Pour plus d'informations, consultez Notifications d'événements Amazon S3.

Lorsque vous créez une tâche Lancer une restauration d'objet S3, les arguments suivants sont disponibles :

ExpirationInDays

Cet argument spécifie combien de temps l'objet S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive reste disponible dans Amazon S3. Les tâches de lancement de restauration d'objet qui ciblent des objets S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive exigent que vous définissiez `ExpirationInDays` sur 1 ou plus.

> **Important**
>
> Ne définissez pas `ExpirationInDays` lors de la création de tâches d'opération Lancer une restauration d'objet S3 qui ciblent des objets de niveau Archive Access ou Deep Archive Access de S3 Intelligent-Tiering. Les objets des niveaux Archive Access de S3 Intelligent-Tiering ne sont pas soumis à l'expiration de la restauration. Par conséquent, spécifier `ExpirationInDays` entraîne un échec de la demande de restauration.

GlacierJobTier

Amazon S3 peut restaurer des objets en utilisant un des trois niveaux de récupération suivants : `EXPEDITED`, `STANDARD` ou `BULK`. Cependant, la fonctionnalité S3 Batch Operations ne prend en charge que les niveaux `STANDARD` de récupération. Pour plus d'informations sur les différences entre les niveaux de récupération, consultez Options de récupération des archives.

Pour plus d'informations sur la tarification de chaque niveau, consultez la section Demandes et récupérations de données dans la page Tarification Amazon S3.

Différences entre la restauration depuis S3 Glacier et S3 Intelligent-Tiering

La restauration de fichiers archivés à partir des classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive diffère de la restauration de fichiers de la classe de stockage S3 Intelligent-Tiering dans les niveaux d'accès Archive ou Deep Archive.

- Lorsque vous effectuez une restauration à partir de S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive, une copie temporaire de l'objet est créée. Amazon S3 supprime cette copie après l'expiration de la valeur que vous avez spécifiée dans

l'argument `ExpirationInDays`. Une fois la copie temporaire supprimée, vous devez soumettre une demande de restauration supplémentaire pour accéder à l'objet.

- Lors de la restauration d'objets S3 Intelligent-Tiering archivés, ne spécifiez pas l'argument `ExpirationInDays`. Lorsque vous restaurez un objet depuis le niveau Archive Access ou Deep Archive Access de S3 Intelligent-Tiering, l'objet repasse au niveau d'accès fréquent de S3 Intelligent-Tiering. Après un minimum de 90 jours consécutifs sans accès, l'objet passe automatiquement au niveau Archive Access. Après un minimum de 180 jours consécutifs sans accès, l'objet passe automatiquement au niveau Deep Archive Access.
- Les tâches d'opérations par lot peuvent fonctionner sur les objets de classe de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive, ou sur les objets des niveaux de stockage Archive Access et Deep Archive Access de S3 Intelligent-Tiering. Les opérations par lot ne peuvent pas fonctionner sur les deux types d'objets archivés dans la même tâche. Pour restaurer des objets des deux types, vous devez créer des tâches Batch Operations.

Chevauchement de restaurations

Si la tâche S3 Initiate Restore Object (Lancer une restauration d'objet S3) essaie de restaurer un objet dont la restauration est déjà en cours, S3 Batch Operations procède comme suit.

La restauration de l'objet aboutit si l'une des conditions suivantes est vraie :

- Comparée à la demande de restauration en cours, la valeur `ExpirationInDays` de cette tâche est la même et sa valeur `GlacierJobTier` est plus rapide.
- La demande de restauration précédente est déjà terminée, et l'objet est actuellement disponible. Dans ce cas, les opérations par lot mettent à jour la date d'expiration de l'objet restauré pour qu'elle corresponde à la valeur `ExpirationInDays` spécifiée dans la demande de restauration en cours.

L'opération de restauration échoue pour l'objet si l'une des conditions suivantes est vraie :

- La demande de restauration en cours n'est pas encore terminée, et la durée de restauration de cette tâche (spécifiée par la valeur `ExpirationInDays`) diffère de la durée de restauration spécifiée dans la demande de restauration en cours.
- Le niveau de restauration de cette tâche (spécifié par la valeur `GlacierJobTier`) est inférieur ou égal au niveau de restauration spécifié dans la demande de restauration en cours.

Limites

Les tâches S3 Initiate Restore Object (Lancer une restauration d'objet S3) présentent les limitations suivantes :

- Vous devez créer la tâche dans la même Région que les objets archivés.
- S3 Batch Operations ne prend pas en charge le niveau d'extraction `EXPEDITED`.

Pour plus d'informations sur la restauration des objets, consultez Restauration d'un objet archivé.

Conservation d'un verrouillage d'objet S3

L'opération de rétention du verrouillage d'objets vous permet d'appliquer des dates de rétention à vos objets en mode gouvernance ou en mode conformité. Ces modes de rétention appliquent différents niveaux de protection. Vous pouvez appliquer l'un ou l'autre des modes de rétention à n'importe quelle version d'objet. Les dates de conservation, comme les blocages légaux, empêchent l'écrasement ou la suppression d'un objet. Amazon S3 stocke la date de fin de conservation spécifiée dans les métadonnées de l'objet et protège la version spécifiée de la version de l'objet jusqu'à l'expiration de la période de conservation.

Vous pouvez utiliser les opérations par lot S3 avec le verrouillage d'objet pour gérer simultanément les dates de conservation de nombreux objets Amazon S3. Vous spécifiez la liste des objets cibles dans votre manifeste et l'envoyez aux tâches d'opérations par lot pour terminer. Pour de plus amples informations, veuillez consulter Verrouillage d'objet S3 the section called “Périodes de rétention”.

Votre tâche d'opérations par lot S3 avec des dates de conservation s'exécute jusqu'à la fin, jusqu'à son annulation ou jusqu'à ce qu'un état d'échec soit atteint. Vous devez utiliser la conservation des opérations par lot S3 et du verrouillage d'objet S3 lorsque vous souhaitez ajouter, modifier ou supprimer la date de conservation pour de nombreux objets avec une seule requête.

Les opérations par lot vérifient que le verrouillage d'objet est activé sur votre compartiment avant de traiter les clés du manifeste. Pour effectuer les opérations et la validation, les opérations par lot ont besoin des autorisations `s3:GetBucketObjectLockConfiguration` et `s3:PutObjectRetention` dans un rôle IAM pour permettre aux opérations par lot d'appeler le verrouillage d'objet en votre nom. Pour de plus amples informations, veuillez consulter the section called “Considérations relatives au verrouillage d’objet”.

Pour de plus amples informations sur l'utilisation de cette opération avec l'API REST, veuillez consulter `S3PutObjectRetention` dans l'opération CreateJob dans la Référence d'API Amazon Simple Storage Service.

Pour un exemple AWS Command Line Interface d'utilisation de cette opération, veuillez consulter the section called “Utiliser les tâches d'opérations par lot avec la rétention du verrouillage des objets”. Pour obtenir un exemple AWS SDK for Java, veuillez consulter the section called “Utiliser les tâches d'opérations par lot avec la rétention du verrouillage des objets”.

Limites et restrictions

- Les opérations par lot S3 n'apportent aucune modification au niveau du compartiment.
- La gestion des versions et le verrouillage d'objet S3 doivent être configurés sur le compartiment où la tâche est effectuée.
- Tous les objets répertoriés dans le manifeste doivent se trouver dans le même compartiment.
- L'opération fonctionne sur la dernière version de l'objet, sauf si une version est explicitement spécifiée dans le manifeste.
- Vous avez besoin d'une autorisation `s3:PutObjectRetention` dans votre rôle IAM pour l'utiliser.
- `s3:GetBucketObjectLockConfiguration`L'autorisation IAM est nécessaire pour confirmer que ce verrouillage d'objet est activé pour le compartiment S3.
- Vous pouvez uniquement prolonger la période de rétention des objets avec des dates de rétention en mode `COMPLIANCE` appliquées, et elle ne peut pas être raccourcie.

Mise en suspens juridique du verrouillage des objets S3

L'opération de mise en suspens juridique du verrouillage des objets vous permet de mettre en place une détention légale sur une version d'objet. À l'instar d'une période de rétention, une mise en suspens juridique empêche une version d'objet d'être remplacée ou supprimée. Pourtant, une détention légale ne possède pas de période de rétention associée et reste en vigueur jusqu'à sa suppression.

Vous pouvez utiliser les opérations par lot S3 avec le verrouillage des objets pour ajouter des mises en suspens juridiques à plusieurs objets Amazon S3 à la fois. Vous pouvez le faire en listant les objets cibles dans votre manifeste et en envoyant cette liste aux opérations par lot. Votre tâche d'opérations par lot S3 avec une conservation du verrouillage d'objet à des fins juridiques s'exécute jusqu'à la fin, jusqu'à son annulation ou jusqu'à ce qu'un état d'échec soit atteint.

Les opérations par lot S3 vérifient que le verrouillage d'objet est activé sur votre compartiment S3 avant de traiter les clés du manifeste. Pour effectuer les opérations d'objet et la validation au niveau du compartiment, S3 Batch Operations nécessite `s3:PutObjectLegalHold` et `s3:GetBucketObjectLockConfiguration` dans un rôle IAM pour lui permettre d'appeler un verrouillage d'objet S3 en votre nom.

Lorsque vous créez la tâche d'opérations par lot S3 pour supprimer la conservation à des fins juridiques, il vous suffit de spécifier Off (Désactivé) comme statut de conservation à des fins juridiques. Pour de plus amples informations, veuillez consulter the section called “Considérations relatives au verrouillage d’objet”.

Pour plus d'informations sur l'utilisation de cette opération avec l'API REST, consultez `S3PutObjectLegalHold` dans l'opération CreateJob dans la référence de l'API Amazon Simple Storage Service.

Pour un exemple d'utilisation de cette opération, consultez Utilisation du AWS SDK pour Java.

Limites et restrictions

- Les opérations par lot S3 n'apportent aucune modification au niveau du compartiment.
- Tous les objets répertoriés dans le manifeste doivent se trouver dans le même compartiment.
- La gestion des versions et le verrouillage d'objet S3 doivent être configurés sur le compartiment où la tâche est effectuée.
- L'opération fonctionne sur la dernière version de l'objet, sauf si une version est explicitement spécifiée dans le manifeste.
- `s3:PutObjectLegalHold`L'autorisation est nécessaire dans votre rôle IAM pour ajouter ou supprimer une conservation à des fins juridiques des objets.
- `s3:GetBucketObjectLockConfiguration`L'autorisation IAM est requise pour confirmer que le verrouillage d'objet S3 est activé pour le compartiment S3.

- Copie d'objets
- AWS Lambda Fonction Invoke
- Replace all object tags (Remplacer toutes les étiquettes des objets)
- Delete all object tags (Supprimer toutes les étiquettes des objets)
- Replace access control list (Remplacer la liste de contrôle d'accès (ACL)
- Restauration d'objets à l'aide d'opérations par lot

- Conservation d'un verrouillage d'objet S3
- Mise en suspens juridique du verrouillage des objets S3
- Réplication d'objets existants via la réplication par lot S3

Gestion des tâches d'opérations par lot S3

Simple Storage Service (Amazon S3) fournit un ensemble d'outils puissants pour vous aider à gérer vos tâches d'opérations par lot S3 après les avoir créées. Cette section décrit les opérations que vous pouvez utiliser pour gérer et suivre vos tâches à l'aide de la AWS Management Console, de la AWS CLI, des kits SDK AWS ou de l'API REST.

Rubriques

- Utilisation de la console Simple Storage Service (Amazon S3) pour gérer vos tâches d'opérations par lot S3
- Liste des tâches
- Affichage des détails de la tâche
- Affectation d'une priorité de tâche

Utilisation de la console Simple Storage Service (Amazon S3) pour gérer vos tâches d'opérations par lot S3

À l'aide de la console, vous pouvez gérer vos tâches d'opérations par lot S3. Par exemple, vous pouvez :

- Afficher les tâches actives et mises en file d'attente
- Modifier la priorité d'une tâche
- Confirmer et exécuter une tâche
- Cloner une tâche
- Annuler une tâche

Pour gérer les opérations par lot via la console

1. Connectez-vous à la AWS Management Console et ouvrez la console Simple Storage Service (Amazon S3) à la page https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Sélectionnez la tâche que vous souhaitez gérer.

Liste des tâches

Vous pouvez récupérer une liste des tâches d'opérations par lot S3. Cette liste inclut les tâches qui ne sont pas encore terminé, ainsi que celles terminées au cours des 90 derniers jours. La liste des tâches inclut des informations pour chacune des tâches telles que l'ID, la description, la priorité, le statut actuel et le nombre de tâches réussies ou ayant échoué. Vous pouvez filtrer la liste de tâches par statut. Lorsque vous procédez à l'extraction d'une liste des tâches via la console, vous pouvez également rechercher vos tâches par description ou ID et les filtrer par Région AWS.

Obtenir une liste des tâches actives et terminées

L'exemple AWS CLI suivant obtient une liste de tâches `Active` et `Complete`.

```
aws s3control list-jobs \
    --region us-west-2 \
    --account-id acct-id \
    --job-statuses '["Active","Complete"]' \
    --max-results 20
```

Pour plus d'informations et des exemples, consultez list-jobs dans la Référence des commandes AWS CLI.

Affichage des détails de la tâche

Si vous souhaitez obtenir d'autres d'informations sur une tâche que celles fournies par la liste des tâches, vous pouvez afficher tous les détails d'une seule tâche. Vous pouvez consulter les détails des tâches qui n'ont pas encore été achevées ou qui ont été achevées au cours des 90 derniers jours. Outre les informations renvoyées dans la liste de tâches, les détails d'une seule tâche incluent d'autres éléments, tels que les suivants :

- Paramètres de fonctionnement
- Détails sur le manifeste
- Informations sur le rapport de fin de tâche (si vous en avez configuré un au moment de créer la tâche)
- Amazon Resource Name (ARN) du rôle d'utilisateur que vous avez attribué pour exécuter la tâche

En affichant les détails d'une tâche individuelle, vous pouvez accéder à la totalité de la configuration d'une tâche. Pour afficher les détails d'une tâche, vous pouvez utiliser la console Amazon S3 ou l'AWS Command Line Interface (AWS CLI).

Obtenir la description d'une tâche d'opérations par lot S3 dans la console Amazon S3

Pour afficher la description d'une tâche d'opérations par lot à l'aide de la console

1. Connectez-vous à la AWS Management Console et ouvrez la console Simple Storage Service (Amazon S3) à la page https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Choisissez l'ID de la tâche spécifique pour en afficher les détails.

Obtenir la description d'une tâche d'opérations par lot S3 dans l'AWS CLI

L'exemple suivant permet d'obtenir la description d'une tâche d'opérations par lot S3 à l'aide de l'AWS CLI. Pour utiliser l'exemple de commande suivant, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control describe-job \
--region us-west-2 \
--account-id acct-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Pour plus d'informations et des exemples, consultez describe-job dans la Référence des commandes AWS CLI.

Affectation d'une priorité de tâche

Vous pouvez attribuer à chaque tâche une priorité numérique, qui peut être n'importe quel entier positif. La fonctionnalité d'opérations par lot S3 classe les tâches par ordre de priorité en fonction de la priorité attribuée. Les tâches avec une priorité élevée (ou une valeur d'entier supérieure pour le paramètre de priorité) sont évaluées en premier. La priorité est déterminée par ordre décroissant. Par exemple, une file d'attente de tâches avec une valeur de priorité 10 se voit accorder la préférence en termes de planification par rapport à une file d'attente de tâches avec une valeur de priorité 1.

Vous pouvez modifier la priorité d'une tâche pendant son exécution. Si vous envoyez une nouvelle tâche avec une priorité supérieure alors qu'une tâche est en cours d'exécution, la tâche avec

la priorité inférieure peut s'interrompre pour permettre à la tâche avec la priorité supérieure de s'exécuter.

La modification de la priorité des tâches n'affecte pas la vitesse de traitement des tâches.

> **Note**
>
> La fonctionnalité d'opérations par lot S3 respecte les priorités de tâche dans la mesure du possible. Bien que les tâches avec des priorités supérieures soient généralement prioritaires sur les tâches avec des priorités inférieures, Simple Storage Service (Amazon S3) ne garantit pas un ordre strict des tâches.

Utilisation de la console S3

Comment mettre à jour la priorité des tâches dans le AWS Management Console

1. Connectez-vous à la AWS Management Console et ouvrez la console Simple Storage Service (Amazon S3) à la page https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Batch Operations (Opérations par lot).
3. Sélectionnez la tâche spécifique que vous souhaitez gérer.
4. Choisissez Actions. Dans la liste déroulante, choisissez Update priority (Mettre à jour la priorité).

Utilisation de la AWS CLI

L'exemple suivant met à jour la priorité de la tâche à l'aide de la AWS CLI. Un nombre plus élevé indique une priorité d'exécution plus élevée.

```
aws s3control update-job-priority \
    --region us-west-2 \
    --account-id acct-id \
    --priority 98 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Utilisation de la AWS SDK for Java

L'exemple suivant met à jour la priorité d'une tâche d'opérations par lot S3 à l'aide du kit AWS SDK for Java.

Pour de plus amples informations sur la priorité d'une tâche, consultez Affectation d'une priorité de tâche.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobPriorityRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobPriority {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobPriority(new UpdateJobPriorityRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withPriority(98));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Suivi de l'état de la tâche et des rapports de fin de tâche

Grâce aux opérations par lot S3, vous pouvez afficher et mettre à jour l'état d'une tâche, ajouter des notifications et effectuer la journalisation, suivre les échecs de tâche et générer des rapports de fin de tâches.

Rubriques

Statuts de la tâche

Après la création et l'exécution d'une tâche, celle-ci passe par une série d'états. Le tableau suivant décrit les états et les transitions possibles entre eux.

État	Description	Transitions
`New`	Une tâche commence à l'état `New` lorsque vous la créez.	Une tâche passe automatiquement au statut `Preparing` quand Amazon S3 commence à traiter l'objet de manifeste.
`Preparing`	Amazon S3 traite l'objet manifeste et d'autres paramètres de tâche pour configurer et exécuter la tâche.	Une tâche passe automatiquement au statut `Ready` une fois qu'Amazon S3 a terminé de traiter l'objet de manifeste et les autres paramètres. Elle

État	Description	Transitions
		peut ensuite exécuter l'opération spécifiée pour les objets répertoriés dans le manifeste. Si la tâche nécessite une confirmation avant l'exécution, ce qui est, par exemple, le cas lors de la création d'une tâche avec la console Amazon S3, celle-ci passe du statut `Preparing` à `Suspended`. Elle reste dans l'état `Suspended` jusqu'à ce que vous confirmiez son exécution.
`Suspended`	La tâche nécessite que vous confirmiez son exécution. Seules les tâches créées avec la console Amazon S3 nécessitent une confirmation. Une tâche créée à l'aide de la console passe à l'état `Suspended` juste après `Preparing`. Une fois que vous confirmez que vous souhaitez exécuter la tâche et que la tâche passe à l'était `Ready`, elle ne revient jamais à l'état `Suspended`.	Une fois que vous confirmez que vous souhaitez exécuter la tâche, son état devient `Ready`.

État	Description	Transitions
`Ready`	Amazon S3 peut commencer à exécuter les opérations demandées pour les objets.	Une tâche passe automatiquement au statut `Active` quand Amazon S3 commence à l'exécuter. La durée pendant laquelle une tâche reste dans l'état `Ready` dépend du niveau de priorité et de la durée des autres tâches qui sont déjà en cours d'exécution.
`Active`	Amazon S3 effectue l'opération demandée pour les objets répertoriés dans le manifeste. Pendant qu'une tâche est en cours`Active`, vous pouvez suivre sa progression à l'aide de la console Amazon S3 ou de l'`DescribeJob` opération via l'API REST ou AWS les SDK. AWS CLI	Une tâche quitte l'état `Active` lorsque plus aucune opération n'est exécutée sur les objets. Cela peut se produire automatiquement, par exemple lorsqu'une tâche aboutit ou échoue. Toutefois, cela peut également être le résultat d'une action utilisateur, telle que l'annulation d'une tâche. L'état dans lequel passe la tâche dépend de la raison de la transition.
`Pausing`	La tâche passe d'un certain état à `Paused`.	Une tâche passe automatiquement à l'état `Paused` quand l'étape `Pausing` est terminée.
`Paused`	Une tâche peut passer à l'état `Paused` si vous soumettez une autre tâche de priorité plus élevée alors que cette tâche est exécutée.	Une tâche `Paused` retourne automatiquement à l'état `Active` lorsque les tâches de priorité plus élevée qui bloquent l'exécution de cette tâche se terminent, échouent ou sont suspendues.

État	Description	Transitions
`Complete`	La tâche a fini d'effectuer l'opération demandée sur tous les objets répertoriés dans le manifeste. L'opération peut avoir abouti ou échoué pour chaque objet. Si vous avez configuré la tâche de sorte à générer un rapport de fin de tâche, ce rapport est disponibl e dès que la tâche passe à l'état `Complete`.	`Complete` est un état final. Une fois qu'une tâche atteint l'état `Complete`, elle ne passe plus à aucun autre état.
`Cancelling`	La tâche passe à l'état `Cancelled` .	Une tâche passe automatiq uement à l'état `Cancelled` quand l'étape `Cancelling` est terminée.
`Cancelled`	Vous avez demandé que la tâche soit annulée, ce qui a été fait par la fonctionn alité d'opérations par lot S3. Cette tâche n'enverra plus de nouvelles demandes à Amazon S3.	`Cancelled` est un état final. Après qu'une tâche atteint l'état `Cancelled` , elle ne passe plus à aucun autre état.
`Failing`	La tâche passe à l'état `Failed`.	Une tâche passe automatiq uement à l'état `Failed` une fois que l'étape `Failing` est terminée.

État	Description	Transitions
`Failed`	La tâche a échoué et n'est plus en cours d'exécution. Pour de plus amples informations sur les échecs de tâche, veuillez consulter Échec de suivi de la tâche.	`Failed` est un état final. Après qu'une tâche atteint l'état `Failed`, elle ne passe plus à aucun autre état.

Mise à jour de l'état

Les exemples suivants AWS CLI et ceux du SDK for Java mettent à jour le statut d'une tâche Batch Operations. Pour en savoir plus sur l'utilisation de la console S3 pour gérer les tâches d'opérations par lot, veuillez consulter Utilisation de la console Simple Storage Service (Amazon S3) pour gérer vos tâches d'opérations par lot S3.

À l'aide du AWS CLI

- Si vous n'avez pas spécifié le paramètre `--no-confirmation-required` dans l'exemple `create-job` précédent, la tâche reste à l'état suspendu jusqu'à ce que vous confirmiez cette tâche en définissant son statut sur `Ready`. Amazon S3 rend ensuite la tâche éligible à l'exécution.

```
aws s3control update-job-status \
    --region us-west-2 \
    --account-id 181572960644 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
    --requested-job-status 'Ready'
```

- Annulez la tâche en définissant son statut sur `Cancelled`.

```
aws s3control update-job-status \
     --region us-west-2 \
     --account-id 181572960644 \
     --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
     --status-update-reason "No longer needed" \
     --requested-job-status Cancelled
```

Utilisation du AWS SDK pour Java

L'exemple suivant met à jour l'état d'une tâche d'opérations par lot S3 à l'aide du kit AWS SDK for Java.

Pour de plus amples informations sur l'état d'une tâche, veuillez consulter Suivi de l'état de la tâche et des rapports de fin de tâche.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobStatus {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withRequestedJobStatus("Ready"));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
```

```
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Notifications et journalisation

En plus de demander des rapports de fin de tâche, vous pouvez également capturer, examiner et auditer l'activité d'opérations par lot à l'aide d AWS CloudTrail. Puisque les opérations par lot utilisent des API Amazon S3 existantes pour exécuter des tâches, ces dernières émettent également les mêmes événements que si vous les appeliez directement. Ainsi, vous pouvez suivre et enregistrer la progression de votre tâche et l'ensemble de ses tâches à l'aide des mêmes outils et processus de notification, de journalisation et d'audit que vous utilisez déjà avec Amazon S3. Pour plus d'informations, consultez les exemples dans les sections suivantes.

> **Note**
>
> Amazon S3 Batch Operations génère à la fois des événements de gestion et de données CloudTrail lors de l'exécution de la tâche. Le volume de ces événements évolue avec le nombre de clés dans le manifeste de chaque tâche. Reportez-vous à la page de CloudTrail tarification pour plus de détails, qui inclut des exemples de la façon dont les prix changent en fonction du nombre de sentiers que vous avez configurés dans votre compte. Pour découvrir comment configurer et journaliser des événements en fonction de vos besoins, veuillez consulter la section Création de votre premier journal d'activité du Guide de l'utilisateur AWS CloudTrail .

Pour de plus amples informations sur les évènements Amazon S3, veuillez consulter Notifications d'événements Amazon S3.

Échec de suivi de la tâche

Si une tâche d'opérations par lot S3 rencontre un problème qui l'empêche de s'exécuter avec succès, comme le fait de ne pas pouvoir lire le manifeste spécifié, elle échoue. Lorsqu'une tâche échoue, elle génère un ou plusieurs codes d'échec ou causes d'échec. La fonctionnalité d'opérations par lot S3 stocke les codes et les causes d'échec avec la tâche afin que vous puissiez les consulter en

demandant les détails de la tâche. Si vous avez demandé un rapport de fin de tâche, ce dernier inclut également les codes et les raisons d'échec.

Pour empêcher les tâches d'exécuter un grand nombre d'opérations infructueuses, Amazon S3 impose un seuil d'échec à chaque tâche d'opérations par lot. Lorsqu'une tâche a exécuté au moins 1 000 tâches, Amazon S3 surveille le taux d'échec. Si, à un moment donné, le taux d'échec (le nombre de tâches ayant échoué par rapport au nombre total de tâches exécutées) dépasse 50 %, la tâche échoue. Lorsqu'une tâche échoue parce qu'elle dépasse ce seul, vous pouvez identifier la cause de cet échec. Il se peut, par exemple, que vous ayez inclus par erreur des objets dans le manifeste qui ne sont pas dans le compartiment spécifié. Après avoir corrigé les erreurs, vous pouvez renvoyer la tâche.

> **Note**
>
> La fonctionnalité d'opérations par lot S3 fonctionne de manière asynchrone et n'exécute pas nécessairement les tâches dans l'ordre d'apparition des objets dans le manifeste. Dès lors, vous ne pouvez pas utiliser l'ordre du manifeste pour déterminer les tâches d'objets qui réussissent et celles qui échouent. Vous pouvez plutôt examiner le rapport d'achèvement de la tâche (si vous en avez demandé un) ou consulter vos journaux d' AWS CloudTrail événements pour déterminer la source des échecs.

Rapports de fin de tâche

Lorsque vous créez une tâche, vous pouvez demander un rapport de fin de tâche. Si la fonctionnalité d'opérations par lot S3 appelle avec succès au moins une tâche, Amazon S3 génère un rapport de fin de tâche après avoir terminé l'exécution des tâches, après avoir échoué ou après avoir été annulé. Vous pouvez configurer le rapport de fin de tâche pour y inclure toutes les tâches ou uniquement celles ayant échoué.

Le rapport de fin de tâche inclut la configuration et le statut des tâches, ainsi que des informations sur chaque tâche dont la clé d'objet et la version, le statut, les codes d'erreur et les descriptions des erreurs. Toutefois, les rapports de fin de tâche fournissent un moyen facile de consulter les résultats de vos tâches dans un format consolidé sans nécessiter de configuration supplémentaire. Les rapports d'achèvement sont chiffrés à l'aide des clés gérées par Amazon S3 (SSE-S3). Pour obtenir un exemple de rapport de fin de tâche, veuillez consulter Exemples : Rapports de fin de tâche d'opérations par lot S3.

Si vous ne configurez pas de rapport d'achèvement, vous pouvez toujours surveiller et auditer votre tâche et ses tâches à l'aide d' CloudTrail Amazon CloudWatch. Pour plus d'informations, consultez la section suivante.

Rubriques

- Examples : Suivi d'une tâche d'opérations par lot S3 dans Amazon EventBridge via AWS CloudTrail
- Exemples : Rapports de fin de tâche d'opérations par lot S3

Examples : Suivi d'une tâche d'opérations par lot S3 dans Amazon EventBridge via AWS CloudTrail

L'activité de la tâche d'opérations par lot Amazon S3 est enregistrée en tant qu'événements dans AWS CloudTrail. Vous pouvez créer une règle personnalisée dans Amazon EventBridge et envoyer ces événements à la ressource de notification cible de votre choix, telle qu'Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. Amazon CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaissent dans chaque console. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur Amazon EventBridge.

Exemples de suivi

- Événements d'opérations par lot S3 enregistrés dans CloudTrail
- Règle EventBridge pour le suivi des événements de tâche d'opérations par lot S3

Événements d'opérations par lot S3 enregistrés dans CloudTrail

Lorsqu'une tâche d'opérations par lot est créée, elle est enregistrée en tant qu'événement `JobCreated` dans CloudTrail. Au fur et à mesure que la tâche s'exécute, elle change d'état pendant le traitement et d'autres événements `JobStatusChanged` sont enregistrés dans CloudTrail. Vous

pouvez afficher ces événements sur la console CloudTrail. Pour de plus amples informations sur CloudTrail, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.

> **Note**
>
> Seuls les événements `status-change` des tâches d'opérations par lot S3 sont enregistrés dans CloudTrail.

Example Événement de fin de tâche d'opérations par lot S3 enregistré par CloudTrail

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-05T18:25:30Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "f907577b-bf3d-4c53-b9ed-8a83a118a554",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123412341234",
    "serviceEventDetails": {
        "jobId": "d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "jobArn": "arn:aws:s3:us-west-2:181572960644:job/
d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "status": "Complete",
        "jobEventId": "b268784cf0a66749f1a05bce259804f5",
        "failureCodes": [],
        "statusChangeReason": []
    }
}
```

Règle EventBridge pour le suivi des événements de tâche d'opérations par lot S3

L'exemple suivant montre comment créer une règle dans Amazon EventBridge pour capturer les événements d'opérations par lot S3 enregistrés par AWS CloudTrail vers une cible de votre choix.

Pour ce faire, créez une règle en suivant toutes les étapes décrites dans Création de règles EventBridge qui réagissent aux événements. Vous collez la stratégie de modèle d'événement personnalisé d'opérations par lot S3 suivante, le cas échéant, et vous choisissez le service cible de votre choix.

Stratégie de modèle d'événement personnalisé d'opérations par lots S3

```
{
    "source": [
        "aws.s3"
    ],
    "detail-type": [
        "AWS Service Event via CloudTrail"
    ],
    "detail": {
        "eventSource": [
            "s3.amazonaws.com"
        ],
        "eventName": [
            "JobCreated",
            "JobStatusChanged"
        ]
    }
}
```

Les exemples suivants sont deux événements d'opérations par lot qui ont été envoyés à Amazon Simple Queue Service (Amazon SQS) à partir d'une règle d'événement EventBridge. Une tâche d'opérations par lot passe par de nombreux états différents pendant son traitement (`New`, `Preparing`, `Active`, etc.), de sorte que vous pouvez vous attendre à recevoir plusieurs messages pour chaque tâche.

Example Exemple d'événement JobCreated

```
{
    "version": "0",
    "id": "51dc8145-541c-5518-2349-56d7dffdf2d8",
```

```
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2020-02-27T15:25:49Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "11112223334444",
            "invokedBy": "s3.amazonaws.com"
        },
        "eventTime": "2020-02-27T15:25:49Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "JobCreated",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "s3.amazonaws.com",
        "userAgent": "s3.amazonaws.com",
        "eventID": "7c38220f-f80b-4239-8b78-2ed867b7d3fa",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
            "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
            "status": "New",
            "jobEventId": "f177ff24f1f097b69768e327038f30ac",
            "failureCodes": [],
            "statusChangeReason": []
        }
    }
}
```

Example Événement de fin de tâche JobStatusChanged

```
{
  "version": "0",
  "id": "c8791abf-2af8-c754-0435-fd869ce25233",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.s3",
  "account": "123456789012",
  "time": "2020-02-27T15:26:42Z",
  "region": "us-east-1",
```

```
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "1111222233334444",
      "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-27T15:26:42Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "eventID": "0238c1f7-c2b0-440b-8dbd-1ed5e5833afb",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
      "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
      "status": "Complete",
      "jobEventId": "51f5ac17dba408301d56cd1b2c8d1e9e",
      "failureCodes": [],
      "statusChangeReason": []
    }
  }
}
```


Exemples : Rapports de fin de tâche d'opérations par lot S3

Lorsque vous créez une tâche d'opérations par lot S3, vous pouvez demander un rapport de fin de tâche pour toutes les tâches ou uniquement pour les tâches qui ont échoué. Tant qu'au moins une tâche a été appelée avec succès, la fonctionnalité d'opérations par lot S3 génère un rapport pour les tâches qui ont été achevées, qui ont échoué ou qui ont été annulées.

Le rapport de fin contient des informations complémentaires pour chaque tâche, y compris le nom et la version des clés d’objets, le statut, les codes d’erreurs et les descriptions des erreurs éventuelles. La description des erreurs pour chaque tâche ayant échoué peut être utilisée pour diagnostiquer les problèmes pendant la création de tâches, par exemple les autorisations.

> **Note**
>
> Les rapports d'achèvement sont toujours chiffrés avec des clés gérées par Amazon S3 (SSE-S3).

Example Fichier de résultats de manifeste de niveau supérieur

Le fichier `manifest.json` de niveau supérieur contient les emplacements de chaque rapport réussi et (si la tâche a présenté des échecs) l'emplacement des rapports ayant échoué, comme montré dans l'exemple suivant.

```
{
    "Format": "Report_CSV_20180820",
    "ReportCreationDate": "2019-04-05T17:48:39.725Z",
    "Results": [
        {
            "TaskExecutionStatus": "succeeded",
            "Bucket": "my-job-reports",
            "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/
results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
        },
        {
            "TaskExecutionStatus": "failed",
            "Bucket": "my-job-reports",
            "MD5Checksum": "22ee037f3515975f7719699e5c416eaa",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
        }
    ],
    "ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HTTPStatusCode,
 ResultMessage"
}
```

Example Rapports des tâches ayant échoué

Les rapports des tâches ayant échoué contiennent les informations suivantes pour toutes les tâches ayant échoué :

- `Bucket`

- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

L'exemple de rapport suivant montre un cas où la AWS Lambda fonction a expiré, entraînant des défaillances dépassant le seuil de défaillance. La mention a alors été apposé `PermanentFailure`.

```
awsexamplebucket1,image_14975,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:21.155Z 2845ca0d-38d9-4c4b-
abcf-379dc749c452 Task timed out after 3.00 seconds""}"
awsexamplebucket1,image_15897,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:29.610Z 2d0a330b-de9b-425f-
b511-29232fde5fe4 Task timed out after 3.00 seconds""}"
awsexamplebucket1,image_14819,,failed,200,PermanentFailure,"Lambda returned function
 error: {""errorMessage"":""2019-04-05T17:35:22.362Z fcf5efde-74d4-4e6d-b37a-
c7f18827f551 Task timed out after 3.00 seconds""}"
awsexamplebucket1,image_15930,,failed,200,PermanentFailure,"Lambda returned function
 error: {""errorMessage"":""2019-04-05T17:35:29.809Z 3dd5b57c-4a4a-48aa-8a35-
cbf027b7957e Task timed out after 3.00 seconds""}"
awsexamplebucket1,image_17644,,failed,200,PermanentFailure,"Lambda
 returned function error: {""errorMessage"":""2019-04-05T17:35:46.025Z
 10a764e4-2b26-4d8c-9056-1e1072b4723f Task timed out after 3.00 seconds""}"
awsexamplebucket1,image_17398,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:44.661Z 1e306352-4c54-4eba-
aee8-4d02f8c0235c Task timed out after 3.00 seconds""}"
```

Example Rapport des tâches réussies

Les rapports sur les tâches réussies contiennent les informations suivantes concernant les tâches réussies :

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`

- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Dans l'exemple suivant, la fonction Lambda a copié avec succès l'objet Amazon S3 dans un autre compartiment. La réponse Amazon S3 retournée est renvoyée à la fonctionnalité d'opérations par lot S3 et est ensuite écrite dans le rapport d'achèvement final.

```
awsexamplebucket1,image_17775,,succeeded,200,,"{u'CopySourceVersionId':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'RequestId': '3ED5852152014362', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'x-amz-copy-source-version-id':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':
 '3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type':
 'application/xml'}}}"
awsexamplebucket1,image_17763,,succeeded,200,,"{u'CopySourceVersionId':
 '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()),
 u'ETag': '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata':
 {'HTTPStatusCode': 200, 'RetryAttempts': 0, 'HostId': 'GiCZNYr8LHd/
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'RequestId':
 '1BC9F5B1B95D7000', 'HTTPHeaders': {'content-length': '234', 'x-amz-id-2':
 'GiCZNYr8LHd/Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'x-
amz-copy-source-version-id': '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', 'server': 'AmazonS3',
 'x-amz-request-id': '1BC9F5B1B95D7000', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT',
 'content-type': 'application/xml'}}}"
awsexamplebucket1,image_17860,,succeeded,200,,"{u'CopySourceVersionId':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 40, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'x-amz-copy-source-version-id':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', 'server': 'AmazonS3', 'x-amz-request-id':
 '8D9CA56A56813DF3', 'date': 'Fri, 05 Apr 2019 17:35:40 GMT', 'content-type':
 'application/xml'}}}"
```

Contrôle de l'accès et étiquetage des tâches à l'aide d'étiquettes

Vous pouvez étiqueter et contrôler l'accès à vos tâches d'opérations par lot S3 en ajoutant des étiquettes. Les étiquettes peuvent être utilisées pour identifier la personne qui est responsable d'une tâche d'opérations par lot. La présence d'étiquettes de tâche peut octroyer ou limiter la capacité d'un utilisateur à annuler une tâche, activer une tâche dans l'état de confirmation ou modifier le niveau de priorité d'une tâche. Vous pouvez créer des tâches avec des étiquettes qui leur sont attachées et ajouter des étiquettes aux tâches une fois qu'elles ont été créées. Chaque étiquette est une paire clé-valeur que vous pouvez ajouter lors de la création de la tâche ou de sa mise à jour ultérieure.

> **Warning**
>
> Les étiquettes de tâche ne doivent pas contenir d'informations confidentielles ou de données personnelles.

Prenons l'exemple d'étiquetage suivant : supposons que vous souhaitez que votre service Finance crée une tâche d'opérations par lot. Vous pouvez écrire une stratégie AWS Identity and Access Management (IAM) qui permet à un utilisateur d'appeler `Department`, à condition que la tâche soit créée avec la valeur `CreateJob` affectée par l'étiquette `Finance`. En outre, vous pouvez attacher cette stratégie à tous les utilisateurs qui sont membres du département des finances.

En continuant avec cet exemple, vous pouvez écrire une stratégie qui permet à un utilisateur de mettre à jour la priorité de n'importe quelle tâche comportant les étiquettes souhaitées, ou d'annuler toute tâche qui les possède. Pour de plus amples informations, veuillez consulter the section called “Contrôle des autorisations”.

Vous pouvez ajouter des identifications à de nouvelles tâches d’opérations par lot S3 lors de leur création ou les ajouter aux tâches existantes.

Notez les restrictions suivantes liées aux étiquettes :

- Vous pouvez associer jusqu'à 50 étiquettes à une tâche, à condition qu'elles aient des clés d'étiquette uniques.
- Une clé d'étiquette peut comporter jusqu'à 128 caractères Unicode et les valeurs d'étiquette peuvent comporter jusqu'à 256 caractères Unicode.
- La clé et les valeurs sont sensibles à la casse.

Pour de plus amples informations sur les restrictions liées aux étiquettes, consultez Restrictions encadrant les étiquettes définies par l'utilisateur dans le Guide de l'utilisateur AWS Billing and Cost Management.

Opérations d'API liées au balisage des tâches d'opérations par lot S3

Amazon S3 prend en charge les opérations d'API suivantes qui sont spécifiques au balisage des tâches d'opérations par lot S3 :

- GetJobTagging – retourne l'ensemble d'étiquettes associé à une tâche d'opérations par lot.
- PutJobTagging – remplace l'ensemble d'étiquettes associé à une tâche. Il existe deux scénarios de gestion d'étiquettes de tâches d'opérations par lot S3 à l'aide de cette API.
  - La tâche n'a pas d'étiquettes – vous pouvez ajouter un ensemble d'étiquettes à une tâche (la tâche n'a pas d'étiquettes antérieures).
  - La tâche a un ensemble d'étiquettes – pour modifier l'ensemble d'étiquettes existant, vous pouvez soit le remplacer entièrement, soit apporter des modifications à ce dernier en le récupérant à l'aide de GetJobTagging, modifier cet ensemble d'étiquettes et utiliser cette action d'API pour remplacer l'ensemble d'étiquettes par celui que vous avez modifié.

    > **Note**
    >
    > Si vous envoyez cette demande avec un ensemble d'étiquettes vide, la fonctionnalité d'opérations par lot S3 supprime l'ensemble d'étiquettes existant de l'objet. Si vous employez cette méthode, vous serez facturé pour une demande de Niveau 1 (PUT). Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
    > Pour supprimer des étiquettes existantes pour votre tâche d'opérations par lot, l'action `DeleteJobTagging` est préférable, car elle permet d'obtenir le même résultat sans frais.

- DeleteJobTagging – supprime l'ensemble d'étiquettes associé à une tâche d'opérations par lot.

Création d'une tâche d'opérations par lot avec des étiquettes de tâche utilisées pour l'étiquetage

Vous pouvez étiqueter et contrôler l'accès à vos tâches d'opérations par lot S3 en ajoutant des étiquettes. Les étiquettes peuvent être utilisées pour identifier la personne qui est responsable d'une tâche d'opérations par lot. Vous pouvez créer des tâches avec des étiquettes qui leur sont

attachées et ajouter des étiquettes aux tâches une fois qu'elles ont été créées. Pour de plus amples informations, veuillez consulter the section called “Utilisation d'étiquettes”.

Utilisation de la AWS CLI

L'exemple suivant avec la AWS CLI permet de créer une tâche `S3PutObjectCopy` d'opérations par lot S3 en utilisant des étiquettes de tâche pour libeller celle-ci.

1. Sélectionnez l'action ou `OPERATION` que la tâche d'opérations par lot doit exécuter et choisissez votre élément `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::destination-bucket"
  }
}
EOF
```

2. Identifiez les `TAGS` de tâche que vous voulez pour la tâche. Dans ce cas, vous appliquez deux étiquettes, `department` et `FiscalYear`, avec les valeurs `Marketing` et `2020` respectivement.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Spécifiez l'élément `MANIFEST` pour la tâche d'opérations par lot.

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "EXAMPLE_S3BatchOperations_CSV_20180820",
    "Fields": [
```

```
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::example-bucket/example_manifest.csv",
    "ETag": "example-5dc7a8bfb90808fc5d546218"
  }
}
EOF
```

4. Configurez l'élément REPORT pour la tâche d'opérations par lot.

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::example-report-bucket",
  "Format": "Example_Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/copy-with-replace-metadata",
  "ReportScope": "AllTasks"
}
EOF
```

5. Exécutez l'action `create-job` pour créer votre tâche d'opérations par lot avec les entrées définies dans les étapes précédentes.

```
aws \
    s3control create-job \
    --account-id 123456789012 \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn arn:aws:iam::123456789012:role/batch-operations-role \
    --tags "${TAGS//$'\n'/}" \
    --client-request-token "$(uuidgen)" \
    --region us-west-2 \
    --description "Copy with Replace Metadata";
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple suivant crée une tâche d'opérations par lot S3 avec des étiquettes à l'aide du kit AWS SDK for Java.

```
public String createJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::example-manifest-bucket/
manifests/10_manifest.csv";
    final String manifestObjectVersionId = "example-5dc7a8bfb90808fc5d546218";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new
 JobManifestSpec().withFormat(JobManifestFormat.S3InventoryReport_CSV_20161130);

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::example-report-bucket";
    final String jobReportPrefix = "example-job-reports";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final String lambdaFunctionArn = "arn:aws:lambda:us-
west-2:123456789012:function:example-function";

    final JobOperation jobOperation = new JobOperation()
            .withLambdaInvoke(new
 LambdaInvokeOperation().withFunctionArn(lambdaFunctionArn));

    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");
```

```
    final String roleArn = "arn:aws:iam::123456789012:role/example-batch-operations-
role";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Test lambda job")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withTags(departmentTag, fiscalYearTag)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Suppression des étiquettes d’une tâche d’opérations par lot S3

Vous pouvez utiliser ces exemples pour supprimer les étiquettes d’une tâche d’opérations par lot.

Utilisation de la AWS CLI

L'exemple suivant supprime les étiquettes d'une tâche d'opérations par lot à l'aide de la AWS CLI.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Supprimer les étiquettes d'une tâche d'opérations par lot

Example

L'exemple suivant supprime les étiquettes d'une tâche d'opérations par lot S3 à l'aide du kit AWS SDK for Java.

```
public void deleteJobTagging(final AWSS3ControlClient awss3ControlClient,
                             final String jobId) {
    final DeleteJobTaggingRequest deleteJobTaggingRequest = new
 DeleteJobTaggingRequest()
            .withJobId(jobId);

    final DeleteJobTaggingResult deleteJobTaggingResult =
                awss3ControlClient.deleteJobTagging(deleteJobTaggingRequest);
}
```

Mise en place d'étiquettes de tâche pour une tâche d’opérations par lot S3 existante

Vous pouvez utiliser PutJobTagging pour ajouter des étiquettes de tâche à vos tâches d'opérations par lot S3 existantes. Pour plus d'informations, consultez les exemples suivants.

Utilisation de la AWS CLI

Voici un exemple d'utilisation de `s3control put-job-tagging` pour ajouter des étiquettes de tâche à une tâche d'opérations par lot S3 à de la AWS CLI.

> **Note**
>
> Si vous envoyez cette demande avec un ensemble d'étiquettes vide, la fonctionnalité d'opérations par lot S3 supprime l'ensemble d'étiquettes existant de l'objet. De plus, si vous utilisez cette méthode, vous êtes facturé pour une demande de Niveau 1 (PUT). Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
> Pour supprimer des étiquettes existantes pour votre tâche d'opérations par lot, l'action `DeleteJobTagging` est préférable, car elle permet d'obtenir le même résultat sans frais.

1. Identifiez les `TAGS` de tâche que vous voulez pour la tâche. Dans ce cas, vous appliquez deux étiquettes, `department` et `FiscalYear`, avec les valeurs `Marketing` et `2020` respectivement.

   ```
   read -d '' TAGS <<EOF
   [
     {
       "Key": "department",
       "Value": "Marketing"
     },
     {
   ```

```
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Exécutez l'action `put-job-tagging` avec les paramètres requis.

```
aws \
    s3control put-job-tagging \
    --account-id 123456789012 \
    --tags "${TAGS//$'\n'/}" \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple suivant montre comment placer les étiquettes d'une tâche d'opérations par lot S3 à l'aide du kit AWS SDK for Java.

```
public void putJobTagging(final AWSS3ControlClient awss3ControlClient,
                          final String jobId) {
    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final PutJobTaggingRequest putJobTaggingRequest = new PutJobTaggingRequest()
            .withJobId(jobId)
            .withTags(departmentTag, fiscalYearTag);

    final PutJobTaggingResult putJobTaggingResult =
 awss3ControlClient.putJobTagging(putJobTaggingRequest);
}
```

Obtention des étiquettes d'une tâche d'opérations par lot S3

Vous pouvez utiliser `GetJobTagging` pour renvoyer les étiquettes d'une tâche d'opérations par lot S3. Pour plus d'informations, consultez les exemples suivants.

Utilisation de la AWS CLI

L'exemple suivant permet d'obtenir les étiquettes d'une tâche d'opérations par lot à l'aide de la AWS CLI.

```
aws \
    s3control get-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1;
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple suivant permet d'obtenir les étiquettes d'une tâche d'opérations par lot S3 à l'aide de la AWS SDK for Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                 final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
            .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
            awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```

Contrôle des autorisations pour les opérations par lot S3 à l'aide d'étiquettes de tâche

Pour faciliter la gestion des tâches d'opérations par lot S3, vous pouvez ajouter des étiquettes de tâche. Les étiquettes de tâche vous permettent de contrôler l'accès à vos tâches d'opérations par lot et d'imposer l'application d'étiquettes lors de la création de toute tâche.

Vous pouvez appliquer jusqu'à 50 étiquettes de tâche à chaque tâche d'opérations par lot. Cela vous permet de définir des stratégies très granulaires limitant l'ensemble d'utilisateurs qui peuvent modifier la tâche. Les étiquettes de tâche peuvent octroyer ou limiter la capacité d'un utilisateur à annuler une tâche, activer une tâche dans l'état de confirmation ou modifier le niveau de priorité d'une tâche.

En outre, vous pouvez exiger l’application d'étiquettes à toutes les nouvelles tâches et spécifier les paires clé-valeur autorisées pour les étiquettes. Vous pouvez exprimer toutes ces conditions en utilisant le même langage de stratégie IAM. Pour plus d'informations, consultez Actions, ressources et clés de condition pour Amazon S3 dans le Service Authorization Reference.

L'exemple suivant montre comment utiliser des étiquettes de tâche d'opérations par lot S3 pour accorder aux utilisateurs l'autorisation de créer et de modifier uniquement les tâches exécutées au sein d'un service spécifique (par exemple, le service Finance ou Conformité). Vous pouvez également affecter des tâches en fonction du stade de développement auquel elles sont liées, notamment l’assurance qualité ou la production.

Dans cet exemple, vous utilisez les balises de tâche S3 Batch Operations dans les politiques AWS Identity and Access Management (IAM) pour autoriser les utilisateurs à créer et à modifier uniquement les tâches exécutées au sein de leur service. Vous affectez des tâches en fonction du stade de développement auquel elles sont liées, telles que l’assurance qualité ou la production.

Cet exemple utilise les services suivants, chacun utilisant les tâches d'opérations par lot de différentes manières :

- Finance
- Conformité
- Business Intelligence
- Ingénierie

Rubriques

Contrôler l'accès en affectant des étiquettes aux utilisateurs et aux ressources

Dans ce scénario, les administrateurs utilisent le contrôle d'accès basé sur les attributs (ABAC). ABAC est une stratégie d'autorisation IAM qui définit les autorisations en attachant des balises aux utilisateurs et AWS aux ressources.

Les utilisateurs et les tâches se voient attribuer l'une des étiquettes de service suivantes :

Clé : Valeur

- `department : Finance`
- `department : Compliance`
- `department : BusinessIntelligence`
- `department : Engineering`

> **Note**
>
> Les clés et valeurs d'étiquette sont sensibles à la casse.

À l'aide de la stratégie de contrôle d'accès ABAC, vous accordez à un utilisateur du service Finance l'autorisation de créer et de gérer des tâches d'opérations par lot S3 au sein de son service en associant l'étiquette `department=Finance` à son utilisateur.

En outre, vous pouvez attacher à l'utilisateur IAM une stratégie gérée qui permet à tout utilisateur de son entreprise de créer ou de modifier des tâches d'opérations par lot S3 au sein des différents services.

La stratégie de cet exemple comprend trois déclarations de stratégie :

- La première déclaration de la stratégie permet à l'utilisateur de créer une tâche d'opérations par lot à condition que la demande de création de tâche comprenne une étiquette de tâche correspondant à son service. Ceci est exprimé à l'aide de la syntaxe `"${aws:PrincipalTag/department}"`, qui est remplacée par l'étiquette du service de l'utilisateur au moment de l'évaluation de la stratégie. La condition est remplie lorsque la valeur fournie pour l'étiquette du service dans la demande (`"aws:RequestTag/department"`) correspond au département de l'utilisateur.
- La deuxième déclaration de la stratégie permet aux utilisateurs de modifier la priorité des tâches ou de mettre à jour le statut d'une tâche, à condition que la tâche mise à jour corresponde au service de l'utilisateur.
- La troisième déclaration permet à un utilisateur de mettre à jour les étiquettes d'une tâche d'opérations par lot à tout moment via une demande `PutJobTagging`, à condition que (1) l'étiquette de son service soit conservée et (2) la tâche qu'il met à jour se trouve dans son département.

```
{
```

```
      "Version": "2012-10-17",
      "Statement": [
            {
                  "Effect": "Allow",
                  "Action": "s3:CreateJob",
                  "Resource": "*",
                  "Condition": {
                        "StringEquals": {
                              "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                  "Effect": "Allow",
                  "Action": [
                        "s3:UpdateJobPriority",
                        "s3:UpdateJobStatus"
            ],
                  "Resource": "*",
                  "Condition": {
                        "StringEquals": {
                              "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                  "Effect": "Allow",
                  "Action": "s3:PutJobTagging",
                  "Resource": "*",
                  "Condition": {
                        "StringEquals": {
                              "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                              "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        }
    ]
}
```

Balisage des tâches d'opérations par lot par étape et application de limites sur la priorité des tâches

Toutes les tâches d'opérations par lot S3 ont une priorité numérique, qu'Amazon S3 utilise pour décider de leur ordre d'exécution. Dans cet exemple, vous limitez la priorité maximale que la plupart des utilisateurs peuvent affecter aux tâches, avec des plages de priorité plus élevées réservées à un ensemble limité d'utilisateurs privilégiés, comme suit :

- Plage de priorité de l'étape assurance qualité (faible) : 1-100
- Plage de priorité de l'étape production (élevée) : 1-300

Pour ce faire, introduisez un nouveau jeu d'étiquettes représentant l'étape de la tâche :

Clé : Valeur

- `stage : QA`
- `stage : Production`

Création et mise à jour de tâches à faible priorité au sein d'un service

Cette stratégie introduit deux nouvelles restrictions sur la création et la mise à jour de tâches d'opérations par lot S3, en plus de la restriction basée sur le service :

- Les utilisateurs peuvent créer ou mettre à jour des tâches dans leur service avec une nouvelle condition qui exige que la tâche inclue l'étiquette `stage=QA`.
- Les utilisateurs peuvent créer ou mettre à jour la priorité d'une tâche jusqu'à une nouvelle priorité maximale de 100.

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
        "Effect": "Allow",
        "Action": "s3:CreateJob",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA"
```

```
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:UpdateJobStatus"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:UpdateJobPriority",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:ResourceTag/stage": "QA"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:PutJobTagging",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department" : "${aws:PrincipalTag/department}",
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA",
                "aws:ResourceTag/stage": "QA"
            }
        }
```

```
    },
    {
        "Effect": "Allow",
        "Action": "s3:GetJobTagging",
        "Resource": "*"
    }
    ]
}
```

Création et mise à jour de tâches hautement prioritaires au sein d'un service

Un petit nombre d'utilisateurs peut avoir besoin de créer des tâches hautement prioritaires dans les services de l’assurance qualité ou de la production. Pour répondre à ce besoin, vous créez une stratégie gérée adaptée à la stratégie de faible priorité décrite dans la section précédente.

Cette stratégie effectue les opérations suivantes :

- Elle permet aux utilisateurs de créer ou de mettre à jour des tâches dans leur service avec l'étiquette `stage=QA` ou `stage=Production`.
- Elle permet aux utilisateurs de créer ou de mettre à jour la priorité d'une tâche jusqu'à un maximum de 300.

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
                "Effect": "Allow",
                "Action": "s3:CreateJob",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
                            "aws:RequestTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
```

```
                        "s3:RequestJobPriority": 300
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": [
                      "s3:UpdateJobStatus"
            ],
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:UpdateJobPriority",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
                            "aws:ResourceTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
                            "s3:RequestJobPriority": 300
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:PutJobTagging",
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
```

```
                        "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                        "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                    "ForAnyValue:StringEquals": {
                        "aws:RequestTag/stage": [
                            "QA",
                            "Production"
                    ],
                        "aws:ResourceTag/stage": [
                            "QA",
                            "Production"
                    ]
                }
            }
        }
    ]
}
```

Gestion du verrouillage des objets S3 à l’aide des opérations par lot S3

Avec le verrouillage d'objets S3, vous pouvez aussi mettre en place une conservation à des fins juridiques sur une version d'objet. À l'instar d'une période de rétention, une mise en suspens juridique empêche une version d'objet d'être remplacée ou supprimée. Pourtant, une détention légale ne possède pas de période de rétention associée et reste en vigueur jusqu'à sa suppression. Pour de plus amples informations, veuillez consulter Mise en suspens juridique du verrouillage des objets S3.

Pour obtenir des informations sur l'utilisation des opérations par lot S3 avec verrouillage d'objets pour ajouter des détentions légales à de nombreux objets Amazon S3 à la fois, consultez les sections suivantes.

Rubriques

- Activation du verrouillage des objets S3 à l’aide des opérations par lot S3
- Définition de la rétention du verrouillage des objets à l’aide des opérations par lot
- Utilisation de la fonctionnalité d'opérations par lot S3 avec le mode de conformité de rétention du verrouillage des objets S3
- Utiliser la fonctionnalité d'opérations par lot S3 avec le mode de gouvernance de rétention du verrouillage des objets S3

- Utilisation de la fonctionnalité d'opérations par lot S3 pour désactiver la mise en suspens juridique du verrouillage des objets S3

Activation du verrouillage des objets S3 à l'aide des opérations par lot S3

Vous pouvez utiliser la fonctionnalité d'opérations par lot S3 avec le verrouillage des objets S3 pour gérer la conservation ou activer simultanément une mise en suspens juridique de nombreux objets Amazon S3. Vous spécifiez la liste des objets cibles dans votre manifeste et l'envoyez aux tâches d'opérations par lot pour terminer. Pour de plus amples informations, veuillez consulter the section called "Rétention du verrouillage d'objet" et the section called "Mise en suspens juridique du verrouillage d'objet".

Les exemples suivants montrent comment créer un rôle IAM avec des autorisations d'opérations par lot S3 et mettre à jour les autorisations de rôle pour créer des tâches qui activent le verrouillage des objets. Dans les exemples, remplacez les valeurs de variable par celles qui correspondent à vos besoins. Vous devez également disposer d'un manifeste CSV identifiant les objets de votre tâche d'opérations par lot S3. Pour plus d'informations, consultez the section called "Spécification d'un manifeste".

À l'aide du AWS CLI

1. Créez un rôle IAM et autorisez les tâches d'opérations par lot S3 à s'exécuter.

   Cette étape est obligatoire pour toutes les tâches d'opérations par lot S3.

```
export AWS_PROFILE='aws-user'

read -d '' bops_trust_policy <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "batchoperations.s3.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
EOF
aws iam create-role --role-name bops-objectlock --assume-role-policy-document
 "${bops_trust_policy}"
```

2. Configurez l'exécution des tâches d'opérations par lot S3 avec la fonctionnalité de verrouillage des objets S3.

   Dans cette étape, vous autorisez le rôle à effectuer les opérations suivantes :

   a. Exécutez la tâche de verrouillage des objets sur le compartiment S3 qui contient les objets cibles sur lesquels vous souhaitez exécuter la fonctionnalité d'opérations par lot.

   b. Lisez le compartiment S3 où se trouvent le fichier CSV manifeste et les objets.

   c. Écrivez les résultats de la tâche d'opérations par lot S3 dans le compartiment de rapports.

```
read -d '' bops_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::{{ManifestBucket}}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{ManifestBucket}}/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
```

```
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{ReportBucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name object-lock-
permissions --policy-document "${bops_permissions}"
```

Utilisation du AWS SDK pour Java

Les exemples suivants montrent comment créer un rôle IAM avec des autorisations pour les opérations par lot S3 et mettre à jour les autorisations de rôle pour créer des tâches qui activent le verrouillage des objets à l'aide du kit AWS SDK for Java. Dans le code, remplacez toutes les valeurs de variable par celles qui correspondent à vos besoins. Vous devez également disposer d'un manifeste CSV identifiant les objets de votre tâche d'opérations par lot S3. Pour de plus amples informations, veuillez consulter the section called “Spécification d'un manifeste”.

Procédez comme suit :

1. Créez un rôle IAM et autorisez les tâches d'opérations par lot S3 à s'exécuter. Cette étape est obligatoire pour toutes les tâches d'opérations par lot S3.
2. Configurez l'exécution des tâches d'opérations par lot S3 avec la fonctionnalité de verrouillage des objets S3.

   Vous autorisez le rôle à effectuer les opérations suivantes :

   1. Exécutez la tâche de verrouillage des objets sur le compartiment S3 qui contient les objets cibles sur lesquels vous souhaitez exécuter la fonctionnalité d'opérations par lot.
   2. Lisez le compartiment S3 où se trouvent le fichier CSV manifeste et les objets.
   3. Écrivez les résultats de la tâche d'opérations par lot S3 dans le compartiment de rapports.

```
public void createObjectLockRole() {
    final String roleName = "bops-object-lock";
```

```
    final String trustPolicy = "{" +
            "  \"Version\": \"2012-10-17\", " +
            "  \"Statement\": [ " +
            "    { " +
            "      \"Effect\": \"Allow\", " +
            "      \"Principal\": { " +
            "        \"Service\": [" +
            "          \"batchoperations.s3.amazonaws.com\"" +
            "        ]" +
            "      }, " +
            "      \"Action\": \"sts:AssumeRole\" " +
            "    } " +
            "  ]" +
            "}";

    final String bopsPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": \"s3:GetBucketObjectLockConfiguration\"," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ManifestBucket\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:GetObject\"," +
            "                \"s3:GetObjectVersion\"," +
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ManifestBucket/*\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObject\"," +
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ReportBucket/*\"" +
```

```
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final CreateRoleRequest createRoleRequest = new CreateRoleRequest()
            .withAssumeRolePolicyDocument(bopsPermissions)
            .withRoleName(roleName);

    final CreateRoleResult createRoleResult = iam.createRole(createRoleRequest);

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bopsPermissions)
            .withPolicyName("bops-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Définition de la rétention du verrouillage des objets à l'aide des opérations par lot

L'exemple suivant permet à la règle de définir la rétention du verrouillage des objets S3 pour vos objets dans le compartiment manifeste.

Vous mettez à jour le rôle afin d'y inclure des autorisations `s3:PutObjectRetention` et de pouvoir exécuter la rétention du verrouillage des objets sur les objets de votre compartiment.

À l'aide du AWS CLI

```
export AWS_PROFILE='aws-user'

read -d '' retention_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::{{ManifestBucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name retention-permissions
 --policy-document "${retention_permissions}"
```

Utilisation du AWS SDK pour Java

```
public void allowPutObjectRetention() {
    final String roleName = "bops-object-lock";

    final String retentionPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ManifestBucket*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(retentionPermissions)
            .withPolicyName("retention-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Utilisation de la fonctionnalité d'opérations par lot S3 avec le mode de conformité de rétention du verrouillage des objets S3

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3 sur vos objets. Cet exemple définit le mode de rétention sur `COMPLIANCE` et `retain until date` sur le 1er janvier 2025. Il crée une tâche qui cible les objets du compartiment manifeste et signale les résultats dans le compartiment de rapports que vous avez identifié.

À l'aide du AWS CLI

Example Définir la conformité des mentions sur plusieurs objets

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-01T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
```

```
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/compliance-objects-bops",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Set compliance retain-until to 1 Jul 2030";
```

Example Prolongez les **`COMPLIANCE`** modes **`retain until date`** jusqu'au 15 janvier 2025

L'exemple suivant étend la `COMPLIANCE` du mode `retain until date` jusqu'au 15 janvier 2025.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-15T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF
```

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/compliance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/compliance-objects-bops",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Extend compliance retention to 15 Jan 2025";
```

Utilisation du AWS SDK pour Java

Example Définissez le mode de conservation sur CONFORMITÉ et la conservation jusqu'au 1er janvier 2025.

```
public String createComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-
manifest.csv";
    final String manifestObjectVersionId = "your-object-version-Id";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date janFirst = format.parse("01/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(janFirst)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
```

```
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Set compliance retain-until to 1 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example Extension de la **`COMPLIANCE`** du mode de **`retain until date`**

L'exemple suivant étend la `COMPLIANCE` du mode `retain until date` jusqu'au 15 janvier 2025.

```
public String createExtendComplianceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/compliance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";
```

```
    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan15th = format.parse("15/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Extend compliance retention to 15 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Utiliser la fonctionnalité d'opérations par lot S3 avec le mode de gouvernance de rétention du verrouillage des objets S3

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3. Elle illustre comment appliquer la gouvernance de rétention du verrouillage des objets S3 avec la date `retain until date` du 30 janvier 2025 sur plusieurs objets. Il crée une tâche

d'opérations par lot qui utilise le compartiment manifeste et signale les résultats dans le compartiment de rapports.

À l'aide du AWS CLI

Example Appliquez la gouvernance de rétention S3 Object Lock à plusieurs objets, la durée de conservation étant fixée au 30 janvier 2025

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-30T00:00:00",
      "Mode":"GOVERNANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucketT",
```

```
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/governance-objects",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Put governance retention";
```

Example Contourner la gouvernance de rétention sur plusieurs objets

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3. Il montre comment contourner la gouvernance de rétention entre plusieurs objets et crée une tâche d'opérations par lot qui utilise le compartiment manifeste et signale les résultats dans le compartiment de rapports.

```
export AWS_PROFILE='aws-user'

read -d '' bypass_governance_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        }
    ]
```

```
}
EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name bypass-governance-
permissions --policy-document "${bypass_governance_permissions}"

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "BypassGovernanceRetention": true,
    "Retention": {
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/governance-objects-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::REPORT_BUCKET",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/bops-governance",
  "ReportScope": "AllTasks"
```

```
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Remove governance retention";
```

Utilisation du AWS SDK pour Java

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3. Il montre comment appliquer la gouvernance de rétention S3 Object Lock `retain until date` d'ici le 30 janvier 2025 à plusieurs objets. Il crée une tâche d'opérations par lot qui utilise le compartiment manifeste et signale les résultats dans le compartiment de rapports.

Example Appliquez la gouvernance de rétention S3 Object Lock à plusieurs objets, la durée de conservation étant fixée au 30 janvier 2025

```
public String createGovernanceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
```

```
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/governance-objects";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan30th = format.parse("30/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
                            .withRetainUntilDate(jan30th)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Put governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example Contourner la gouvernance de rétention sur plusieurs objets

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3. Il montre comment contourner la gouvernance de rétention entre plusieurs objets et crée une tâche d'opérations par lot qui utilise le compartiment manifeste et signale les résultats dans le compartiment de rapports.

```
public void allowBypassGovernance() {
    final String roleName = "bops-object-lock";

    final String bypassGovernancePermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:BypassGovernanceRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ManifestBucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bypassGovernancePermissions)
            .withPolicyName("bypass-governance-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
public String createRemoveGovernanceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/governance-objects-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";
```

```
    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/bops-governance";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Remove governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
```

```
}
```

Utilisation de la fonctionnalité d'opérations par lot S3 pour désactiver la mise en suspens juridique du verrouillage des objets S3

L'exemple suivant s'appuie sur les exemples précédents de création d'une stratégie d'approbation et de définition des autorisations de configuration pour les opérations par lot S3 et le verrouillage des objets S3. Il montre comment désactiver la mise en suspens juridique du verrouillage des objets sur les objets à l'aide de la fonctionnalité d'opérations par lot.

L'exemple met d'abord à jour le rôle afin d'accorder des autorisations `s3:PutObjectLegalHold`, crée une tâche d'opérations par lot qui désactive (supprime) la mise en suspens juridique des objets identifiés dans le manifeste, puis établit un rapport à ce sujet.

À l'aide du AWS CLI

Example Met à jour le rôle pour accorder des autorisations `s3:PutObjectLegalHold`

```
export AWS_PROFILE='aws-user'

read -d '' legal_hold_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectLegalHold"
            ],
            "Resource": [
                "arn:aws:s3:::ManifestBucket/*"
            ]
        }
    ]

EOF

aws iam put-role-policy --role-name bops-objectlock --policy-name legal-hold-
permissions --policy-document "${legal_hold_permissions}"
```

Example Désactiver la mise en suspens juridique

L'exemple suivant désactive la mise en suspens juridique.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/bops-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectLegalHold": {
    "LegalHold": {
      "Status":"OFF"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::ManifestBucket/legalhold-object-manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/legalhold-objects-bops",
  "ReportScope": "AllTasks"
}
EOF
```

```
aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Turn off legal hold";
```

Utilisation du AWS SDK pour Java

Example Met à jour le rôle pour accorder des autorisations **`s3:PutObjectLegalHold`**

```
public void allowPutObjectLegalHold() {
    final String roleName = "bops-object-lock";

    final String legalHoldPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectLegalHold\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::ManifestBucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(legalHoldPermissions)
            .withPolicyName("legal-hold-permissions")
            .withRoleName(roleName);
```

```
    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Example Désactiver la mise en suspens juridique

Utilisez l'exemple ci-dessous si vous souhaitez désactiver la mise en suspens juridique.

```
public String createLegalHoldOffJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::ManifestBucket/legalhold-object-
manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::ReportBucket";
    final String jobReportPrefix = "reports/legalhold-objects-bops";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectLegalHold(new S3SetObjectLegalHoldOperation()
                    .withLegalHold(new S3ObjectLockLegalHold()
                            .withStatus(S3ObjectLockLegalHoldStatus.OFF)));

    final String roleArn = "arn:aws:iam::123456789012:role/bops-object-lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;
```

```
    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Turn off legal hold")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Tutoriel des opérations par lots S3

Le didacticiel suivant présente les procédures complètes des tâches de certaines opérations par lots.

- Tutoriel : Transcodage par lots de vidéos avec S3 Batch Operations, et AWS LambdaAWS Elemental MediaConvert

Surveillance d'Amazon S3

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances d'Amazon S3 et de vos AWS solutions. Nous vous recommandons de collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de pouvoir corriger plus facilement une panne multipoint, le cas échéant. Avant de commencer à surveiller Amazon S3, créez un plan de surveillance qui comprend les réponses aux questions suivantes :

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

Pour en savoir plus sur la journalisation et la surveillance dans Amazon S3, consultez les rubriques suivantes.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

- Outils de surveillance
- Options de journalisation pour Amazon S3
- Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail
- Enregistrement de demandes avec journalisation des accès au serveur
- Surveillance des métriques avec Amazon CloudWatch
- Notifications d'événements Amazon S3

Outils de surveillance

AWS fournit différents outils que vous pouvez utiliser pour surveiller Amazon S3. Vous pouvez configurer certains outils pour qu'ils effectuent la supervision automatiquement, tandis que d'autres nécessitent une intervention manuelle. Nous vous recommandons d'automatiser le plus possible les tâches de supervision.

Outils de surveillance automatique

Vous pouvez utiliser les outils de surveillance automatique pour surveiller Amazon S3 et signaler un problème éventuel :

- Amazon CloudWatch Alarms — Surveillez une seule métrique sur une période que vous spécifiez et effectuez une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. L'action est une notification envoyée à une rubrique Amazon Simple Notification Service (Amazon SNS) ou à une politique Amazon EC2 Auto Scaling. CloudWatch les alarmes n'appellent pas d'actions simplement parce qu'elles sont dans un état particulier. L'état doit avoir changé et avoir été maintenu pendant un nombre de périodes spécifié. Pour plus d’informations, consultez Surveillance des métriques avec Amazon CloudWatch.
- AWS CloudTrail Surveillance des journaux : partagez les fichiers journaux entre les comptes, surveillez les fichiers CloudTrail CloudWatch journaux en temps réel en les envoyant à Logs, écrivez des applications de traitement des journaux en Java et vérifiez que vos fichiers journaux n'ont pas changé après leur livraison par CloudTrail. Pour plus d’informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.

Outils de surveillance manuelle

Un autre aspect important de la surveillance d'Amazon S3 consiste à surveiller manuellement les éléments non couverts par les CloudWatch alarmes. Les AWS Management Console tableaux de bord Amazon S3 CloudWatch Trusted Advisor,,, et d'autres fournissent une at-a-glance vue de l'état de votre AWS environnement. Vous pouvez activer la journalisation des accès au serveur pour suivre les demandes d'accès à votre compartiment. Chaque enregistrement de journal d'accès fournit des informations détaillées sur une demande d'accès donnée (demandeur, nom du compartiment, heure de la demande, action associée à la demande, état de la réponse et code d'erreur, le cas échéant). Pour plus d’informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.

- Le tableau de bord Amazon S3 affiche les éléments suivants :
  - Vos compartiments et les objets et propriétés qu'ils contiennent
- La page CloudWatch d'accueil affiche les informations suivantes :
  - Alarmes et statuts en cours
  - Graphiques des alarmes et des ressources
  - Statut d’intégrité du service

  En outre, vous pouvez utiliser CloudWatch pour effectuer les opérations suivantes :
  - Créer des tableaux de bord personnalisés pour surveiller les services de votre choix
  - Données de métriques de graphiques pour résoudre les problèmes et découvrir les tendances.
  - Recherchez et parcourez tous les indicateurs de vos AWS ressources.
  - Créer et Modifier des alarmes pour être informé des problèmes.
- AWS Trusted Advisor peut vous aider à surveiller vos AWS ressources afin d'améliorer les performances, la fiabilité, la sécurité et la rentabilité. Quatre contrôles Trusted Advisor sont disponibles pour tous les utilisateurs. Plus de 50 contrôles sont disponibles pour les utilisateurs avec un plan de support Business ou Enterprise. Pour plus d’informations, consultez AWS Trusted Advisor.

  Trusted Advisor possède les vérifications suivantes relatives à Amazon S3 :
  - Vérification de la configuration de journalisation des compartiments Amazon S3.
  - Vérifications de sécurité pour les compartiments Amazon S3 dont les autorisations permettent un libre accès.
  - Vérifications de la tolérance aux pannes pour les compartiments Amazon S3 pour lesquels la gestion des versions est désactivée ou suspendue.

Options de journalisation pour Amazon S3

Vous pouvez enregistrer les actions entreprises par les utilisateurs, les rôles ou Services AWS sur les ressources Amazon S3 et conserver des enregistrements de journal à des fins d'audit et de conformité. Pour ce faire, vous pouvez utiliser la journalisation d'accès au serveur, la journalisation AWS CloudTrail ou une combinaison des deux. Nous vous recommandons de l'utiliser CloudTrail pour consigner les actions au niveau du bucket et au niveau de l'objet pour vos ressources Amazon S3. Pour plus d'informations, consultez les sections suivantes :

- Enregistrement de demandes avec journalisation des accès au serveur

- Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail

Le tableau suivant répertorie les principales propriétés des CloudTrail journaux et des journaux d'accès au serveur Amazon S3. Pour vous assurer que cela CloudTrail répond à vos exigences de sécurité, consultez le tableau et les notes.

Propriétés des journaux	AWS CloudTrail	Journaux du serveur Amazon S3
Peut être transféré vers d'autres systèmes (Amazon CloudWatch Logs, Amazon CloudWatch Events)	Oui	Non
Acheminement des journaux vers plusieurs destinations (par exemple, envoi des mêmes journaux vers deux compartiments différents).	Oui	Non
Activation des journaux sur un sous-ensemble d'objets (préfixe)	Oui	Non
Fourniture de journaux inter-comptes (compartiments cible et source appartenant à des comptes différents)	Oui	Non
Validation de l'intégrité du fichier journal en utilisant la signature numérique ou le hachage.	Oui	Non
Choix ou défaut de chiffrement pour les fichiers journaux.	Oui	Non

Propriétés des journaux	AWS CloudTrail	Journaux du serveur Amazon S3
Opérations d'objet (à l'aide des API Amazon S3)	Oui	Oui
Opérations de compartiment (à l'aide des API Amazon S3)	Oui	Oui
Interface utilisateur pouvant faire l'objet d'une recherche sur les journaux	Oui	Non
Champs pour les paramètres de verrouillage des objets Object Lock, propriétés de sélection Amazon S3 Select pour les enregistrements de journaux	Oui	Non
Champs pour `Object Size`, `Total Time`, `Turn-Around Time` et `HTTP Referer` pour les enregistrements de journaux	Non	Oui
Restaurations, expirations et transitions de cycle de vie	Non	Oui
Consignation des clés dans une opération d'effacement par batchs	Non	Oui
Échecs d'authentification[1]	Non	Oui
Comptes où les journaux ont été livrés	Propriétaire du compartiment[2] et demandeur	Propriétaire du compartiment uniquement
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs

Propriétés des journaux	AWS CloudTrail	Journaux du serveur Amazon S3
Prix	Les événements de gestion (première livraison) sont gratuits ; les événements de données sont payants, en plus du stockage des journaux.	Aucun autre coût en plus du stockage des journaux
Rapidité de fourniture des journaux	Événements de données toutes les 5 minutes ; événements de gestion toutes les 15 minutes	Sous quelques heures
Format des journaux	JSON	Fichier journal avec enregistrements séparés par des espaces, délimités par de nouvelles lignes

Remarques

1. CloudTrail ne fournit pas de journaux pour les demandes qui échouent à l'authentification (pour lesquelles les informations d'identification fournies ne sont pas valides). Cependant, sont fournis les journaux pour les requêtes dans lesquelles l'autorisation a échoué (`AccessDenied`) et les demandes formulées par des utilisateurs anonymes.
2. Le propriétaire du compartiment S3 reçoit CloudTrail des journaux lorsque le compte ne dispose pas d'un accès complet à l'objet figurant dans la demande. Pour plus d'informations, consultez Actions au niveau de l'objet Amazon S3 dans des scénarios entre comptes.
3. S3 ne prend pas en charge la livraison de CloudTrail journaux ou de journaux d'accès au serveur au demandeur ou au propriétaire du compartiment pour les demandes de point de terminaison VPC lorsque la politique de point de terminaison du VPC les refuse ou pour les demandes qui échouent avant que la politique VPC ne soit évaluée.

Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail

Amazon S3 est intégré à AWS CloudTrailun service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture tous les appels d'API pour Amazon S3 sous forme d'événements. Les appels capturés incluent des appels provenant de la console Amazon S3 et des appels de code vers les opérations de l'API Amazon S3. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à Amazon S3, l'adresse IP à partir de laquelle la demande a été faite, la date à laquelle elle a été faite et des informations supplémentaires.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur de l'IAM Identity Center.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre Service AWS.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section Utilisation de l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou CloudTrailLake.

CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous pouvez créer un parcours à région unique ou multirégionale à l'aide du. AWS CLI Il est recommandé de créer

un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un parcours à région unique, vous ne pouvez voir que les événements enregistrés dans le parcours. Région AWS Pour plus d'informations sur les sentiers, consultez les sections Création d'un sentier pour votre organisation Compte AWS et Création d'un sentier pour une organisation dans le guide de AWS CloudTrail l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours dans votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification. Pour obtenir des informations sur la tarification Amazon S3, consultez Tarification Amazon S3.

CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter des requêtes SQL sur vos événements. CloudTrail Lake convertit les événements existants au format JSON basé sur les lignes au format Apache ORC. ORC est un format de stockage en colonnes qui est optimisé pour une récupération rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des sélecteurs d'événements avancés. Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section Travailler avec AWS CloudTrail Lake dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'option de tarification que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

Vous pouvez stocker vos fichiers journaux dans votre compartiment aussi longtemps que vous le souhaitez, ou bien définir des règles de cycle de vie Amazon S3 afin de les archiver ou de les supprimer automatiquement. Par défaut, vos fichiers journaux sont chiffrés à l'aide du chiffrement côté serveur (SSE) d'Amazon S3.

Utilisation de CloudTrail journaux avec les journaux d'accès et CloudWatch les journaux d'accès au serveur Amazon S3

AWS CloudTrail les journaux fournissent un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans Amazon S3, tandis que les journaux d'accès au serveur Amazon S3 fournissent des enregistrements détaillés des demandes adressées à un compartiment S3. Pour plus d'informations sur le fonctionnement des différents journaux ainsi que sur leurs propriétés, performances et coûts, consultez the section called “Options de journalisation”.

Vous pouvez utiliser AWS CloudTrail les journaux avec les journaux d'accès au serveur pour Amazon S3. CloudTrail les journaux vous fournissent un suivi détaillé des API pour les opérations au niveau du bucket et au niveau de l'objet d'Amazon S3. Les journaux d'accès au serveur pour Amazon S3 vous offrent une visibilité sur les opérations au niveau des objets sur vos données dans Amazon S3. Pour plus d'informations sur les journaux d'accès au serveur, consultez Enregistrement de demandes avec journalisation des accès au serveur.

Vous pouvez également utiliser CloudTrail les journaux avec Amazon CloudWatch pour Amazon S3. CloudTrail l'intégration avec CloudWatch Logs fournit l'activité de l'API au niveau du compartiment S3 capturée par CloudTrail un flux de CloudWatch journaux du groupe de CloudWatch journaux que vous spécifiez. Vous pouvez créer des CloudWatch alarmes pour surveiller une activité d'API spécifique et recevoir des notifications par e-mail lorsque l'activité d'API spécifique se produit. Pour plus d'informations sur les CloudWatch alarmes destinées à surveiller une activité d'API spécifique, consultez le guide de AWS CloudTrail l'utilisateur. Pour plus d'informations sur l'utilisation CloudWatch avec Amazon S3, consultezSurveillance des métriques avec Amazon CloudWatch.

> **Note**
>
> S3 ne prend pas en charge la remise de CloudTrail journaux au demandeur ou au propriétaire du compartiment pour les demandes de point de terminaison VPC lorsque la politique de point de terminaison du VPC les refuse.

CloudTrail suivi avec les appels d'API SOAP Amazon S3

CloudTrail suit les appels d'API SOAP Amazon S3. La prise en charge de SOAP par Amazon S3 via HTTP est obsolète, mais continue d'être disponible via HTTPS. Pour plus d’informations sur la prise en charge de SOAP par Amazon S3, consultez Annexe A : Utilisation de l'API SOAP.

> **Important**
>
> Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Actions SOAP Amazon S3 suivies par CloudTrail journalisation

Nom d'API SOAP	Nom de l'événement API utilisé dans le CloudTrail journal
ListAllMyBuckets	`ListBuckets`
CreateBucket	`CreateBucket`
DeleteBucket	`DeleteBucket`
GetBucketAccessControlPolicy	`GetBucketAcl`
SetBucketAccessControlPolicy	`PutBucketAcl`
GetBucketLoggingStatus	`GetBucketLogging`
SetBucketLoggingStatus	`PutBucketLogging`

Pour plus d'informations sur Amazon S3 CloudTrail et Amazon S3, consultez les rubriques suivantes :

Rubriques

- CloudTrail Événements Amazon S3
- CloudTrail entrées de fichiers journaux pour Amazon S3 et S3 on Outposts
- Activation de la journalisation des CloudTrail événements pour les compartiments et les objets S3
- Identification des demandes Amazon S3 à l'aide CloudTrail

CloudTrail Événements Amazon S3

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Cette section fournit des informations sur les événements auxquels S3 se connecte CloudTrail.

Événements liés aux données Amazon S3 dans CloudTrail

Les événements de données fournissent des informations sur les opérations de ressources effectuées sur ou dans une ressource (par exemple, lecture ou écriture de données dans un objet Amazon S3). Ils sont également connus sous le nom opérations de plans de données. Les événements de données sont souvent des activités dont le volume est élevé. Par défaut, CloudTrail n'enregistre pas les événements liés aux données. L'historique des CloudTrail événements n'enregistre pas les événements liés aux données.

Des frais supplémentaires s’appliquent pour les événements de données. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

Vous pouvez enregistrer les événements de données pour les types de ressources Amazon S3 à l'aide de la CloudTrail console ou des opérations CloudTrail d'API. AWS CLI Pour plus d'informations sur la façon de consigner les événements liés aux données, consultez les sections Consignation des événements liés aux données avec le AWS Management Console et Enregistrement des événements liés aux données avec le AWS Command Line Interface dans le Guide de AWS CloudTrail l'utilisateur.

Le tableau suivant répertorie les types de ressources Amazon S3 pour lesquels vous pouvez enregistrer des événements de données. La colonne Type d'événement de données (console) indique la valeur à choisir dans la liste des types d'événements de données de la CloudTrail console.

La colonne de valeur resources.type indique la **`resources.type`** valeur que vous devez spécifier lors de la configuration de sélecteurs d'événements avancés à l'aide des API or. AWS CLI CloudTrail La CloudTrail colonne Data APIs logged to indique les appels d'API enregistrés CloudTrail pour le type de ressource.

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
S3	`AWS::S3::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • DeleteObjects • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • GetObjectTorrent • HeadObject • ListMultipartUploads • ListObjectVersions • ListObjects • ListParts • PutObject • PutObjectAcl • PutObjectLegalHold • PutObjectRetention • PutObjectTagging

Type d’événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
		• RestoreObject • SelectObjectContent • UploadPart • UploadPartCopy

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
Points d'accès S3	`AWS::S3::Access Point`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (copies dans une même Région uniquement) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign

Type d’événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
		• PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • UploadPartCopy (copies dans une même Région uniquement)

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
S3 Object Lambda	`AWS::S3ObjectLambda::AccessPoint`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (copies dans une même Région uniquement) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
S3 Outposts	`AWS::S3Outposts::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (copies dans une même Région uniquement) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Vous pouvez configurer des sélecteurs d'événements avancés pour filtrer les `eventName`, `readOnly`, et `resources.ARN` des champs pour enregistrer uniquement les événements importants pour vous. Pour plus d'informations sur ces champs, consultez AdvancedFieldSelectorla référence de l'AWS CloudTrail API.

Événements de gestion Amazon S3 dans CloudTrail

Amazon S3 enregistre toutes les opérations du plan de contrôle en tant qu'événements de gestion. Pour plus d'informations sur les opérations de l'API S3, consultez le manuel Amazon S3 API Reference.

Comment CloudTrail capture les demandes adressées à Amazon S3

Par défaut, CloudTrail enregistre les appels d'API au niveau du compartiment S3 effectués au cours des 90 derniers jours, mais pas les demandes de journalisation adressées aux objets. Les appels au niveau des compartiments comprennent des événements tels que `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy` et ainsi de suite. Vous pouvez voir les événements au niveau du bucket sur la CloudTrail console. Cependant, vous ne pouvez pas y consulter les événements de données (appels au niveau des objets Amazon S3). Vous devez les analyser ou les consulter dans les journaux. CloudTrail

Actions au niveau du compte Amazon S3 suivies par journalisation CloudTrail

CloudTrail enregistre les actions au niveau du compte. Les enregistrements Amazon S3 sont écrits avec d'autres Service AWS enregistrements dans un fichier journal. CloudTrail détermine à quel moment créer et écrire dans un nouveau fichier en fonction d'une période et de la taille du fichier.

Les tableaux de cette section répertorient les actions au niveau du compte Amazon S3 prises en charge pour la connexion par. CloudTrail

Les actions d'API au niveau du compte Amazon S3 suivies par CloudTrail journalisation apparaissent sous les noms d'événements suivants. Les noms des CloudTrail événements sont différents du nom de l'action de l'API. Par exemple, DeletePublicAccessBlock est DeleteAccountPublicAccessBlock.

- DeleteAccountPublicAccessBlock
- GetAccountPublicAccessBlock
- PutAccountPublicAccessBlock

Actions au niveau du compartiment Amazon S3 suivies par journalisation CloudTrail

Par défaut, CloudTrail enregistre les actions au niveau du compartiment pour les compartiments à usage général. Les enregistrements Amazon S3 sont écrits avec d'autres enregistrements AWS de service dans un fichier journal. CloudTrail détermine à quel moment créer et écrire dans un nouveau fichier en fonction d'une période et de la taille du fichier.

Cette section répertorie les actions au niveau du compartiment Amazon S3 prises en charge pour la journalisation. CloudTrail

Les actions d'API au niveau du bucket Amazon S3 suivies par CloudTrail journalisation apparaissent sous les noms d'événements suivants. Dans certains cas, le nom de l' CloudTrail événement est

différent du nom de l'action de l'API. Par exemple, `PutBucketLifecycleConfiguration` est `PutBucketLifecycle`.

- CreateBucket
- DeleteBucket
- DeleteBucketAnalyticsConfiguration
- DeleteBucketCors
- DeleteBucketEncryption
- DeleteBucketIntelligentTieringConfiguration
- DeleteBucketInventoryConfiguration
- DeleteBucketLifecycle
- DeleteBucketMetricsConfiguration
- DeleteBucketOwnershipControls
- DeleteBucketPolicy
- DeleteBucketPublicAccessBlock
- DeleteBucketReplication
- DeleteBucketTagging
- GetAccelerateConfiguration
- GetBucketAcl
- GetBucketAnalyticsConfiguration
- GetBucketCors
- GetBucketEncryption
- GetBucketIntelligentTieringConfiguration
- GetBucketInventoryConfiguration
- GetBucketLifecycle
- GetBucketLocation
- GetBucketLogging
- GetBucketMetricsConfiguration
- GetBucketNotification
- GetBucketObjectLockConfiguration

- GetBucketOwnershipControls
- GetBucketPolicy
- GetBucketPolicyStatus
- GetBucketPublicAccessBlock
- GetBucketReplication
- GetBucketRequestPayment
- GetBucketTagging
- GetBucketVersioning
- GetBucketWebsite
- HeadBucket
- ListBuckets
- PutAccelerateConfiguration
- PutBucketAcl
- PutBucketAnalyticsConfiguration
- PutBucketCors
- PutBucketEncryption
- PutBucketIntelligentTieringConfiguration
- PutBucketInventoryConfiguration
- PutBucketLifecycle
- PutBucketLogging
- PutBucketMetricsConfiguration
- PutBucketNotification
- PutBucketObjectLockConfiguration
- PutBucketOwnershipControls
- PutBucketPolicy
- PutBucketPublicAccessBlock
- PutBucketReplication
- PutBucketRequestPayment
- PutBucketTagging
- PutBucketVersioning

- PutBucketWebsite

En plus de ces opérations d'API, vous pouvez également utiliser l'action au niveau de l'objet OPTIONS object (objet). Cette action est traitée comme une action au niveau du compartiment dans la CloudTrail journalisation, car elle vérifie la configuration CORS d'un compartiment.

Actions au niveau du bucket S3 Express One Zone (point de terminaison d'API régional) suivies par journalisation CloudTrail

Par défaut, CloudTrail enregistre les actions au niveau du compartiment pour les compartiments de répertoire en tant qu'événements de gestion. Les événements CloudTrail de gestion `eventsource` pour S3 Express One Zone sont`s3express.amazonaws.com`.

> **Note**
>
> Pour S3 Express One Zone, la CloudTrail journalisation des opérations d'API du point de terminaison zonal (au niveau de l'objet ou du plan de données) (par exemple, `PutObject` ou`GetObject`) n'est pas prise en charge.

Les opérations suivantes de l'API de point de terminaison régional sont enregistrées sur CloudTrail.

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- PutBucketPolicy
- ListDirectoryBuckets

Pour plus d'informations, consultez Bonnes pratiques de sécurité pour S3 Express One Zone.

Actions au niveau de l'objet Amazon S3 dans des scénarios entre comptes

Vous trouverez ci-dessous des cas d'utilisation particuliers impliquant les appels d'API au niveau de l'objet dans des scénarios entre comptes et la manière dont les CloudTrail journaux sont signalés. CloudTrail fournit les journaux au demandeur (le compte qui a effectué l'appel d'API), sauf dans

certains cas de refus d'accès où les entrées du journal sont expurgées ou omises. Lorsque vous configurez l'accès entre comptes, pensez aux exemples de cette section.

> **Note**
>
> Les exemples supposent que CloudTrail les journaux sont correctement configurés.

Exemple 1 : CloudTrail fournit des journaux au propriétaire du compartiment

CloudTrail fournit des journaux au propriétaire du compartiment même si celui-ci n'est pas autorisé à effectuer la même opération d'API d'objet. Envisagez le scénario entre comptes suivant :

- Le compte A est le propriétaire du compartiment.
- Le compte B (le demandeur) tente d'accéder à un objet de ce compartiment.
- Le compte C est propriétaire de l'objet. Le compte C peut ou non être le même compte que le compte A.

> **Note**
>
> CloudTrail fournit toujours des journaux d'API au niveau de l'objet au demandeur (compte B). En outre, fournit CloudTrail également les mêmes journaux au propriétaire du bucket (compte A) même si celui-ci n'est pas propriétaire de l'objet (compte C) ou n'est pas autorisé à effectuer les mêmes opérations d'API sur cet objet.

Exemple 2 : CloudTrail ne multiplie pas les adresses e-mail utilisées pour définir les ACL des objets

Envisagez le scénario entre comptes suivant :

- Le compte A est le propriétaire du compartiment.
- Le compte B (le demandeur) envoie une demande pour définir un octroi de liste ACL d'objet en utilisant une adresse e-mail. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL).

Le demandeur obtient les journaux ainsi que les informations de messagerie. Toutefois, le propriétaire du compartiment, s'il est éligible pour recevoir des journaux, comme dans l'exemple, obtient que le CloudTrail journal signale l'événement. Cependant, le propriétaire du compartiment

n'obtient pas les informations de configuration de liste ACL, notamment l'adresse e-mail du bénéficiaire et l'octroi. La seule information transmise par le journal au propriétaire du compartiment est qu'un appel d'API de liste ACL a été passé par le compte B.

CloudTrail entrées de fichiers journaux pour Amazon S3 et S3 on Outposts

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. L'état du chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour le téléchargement de nouveaux objets est disponible dans AWS CloudTrail les journaux, S3 Inventory, S3 Storage Lens, la console Amazon S3 et sous forme d'en-tête de réponse d'API Amazon S3 supplémentaire dans les AWS SDK AWS Command Line Interface et. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération d'API demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics. Les événements n'apparaissent donc pas dans un ordre spécifique.

Pour plus d'informations, consultez les exemples suivants.

Rubriques

- Exemple : entrée de fichier CloudTrail journal pour Amazon S3
- Exemple : entrées du fichier journal Amazon S3 sur Outposts

Exemple : entrée de fichier CloudTrail journal pour Amazon S3

L'exemple suivant montre une entrée de CloudTrail journal illustrant le GETservice et PutBucketAcllesGetBucketVersioningactions.

```
{
    "Records": [
    {
```

```
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2019-02-01T03:18:19Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "ListBuckets",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[]",
        "requestParameters": {
            "host": [
                "s3.us-west-2.amazonaws.com"
            ]
        },
        "responseElements": null,
        "additionalEventData": {
            "SignatureVersion": "SigV2",
            "AuthenticationMethod": "QueryString",
            "aclRequired": "Yes"
    },
        "requestID": "47B8E8D397DCE7A6",
        "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
        "eventType": "AwsApiCall",
        "recipientAccountId": "444455556666",
        "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
       "eventVersion": "1.03",
       "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
            "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:22:33Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "PutBucketAcl",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "bucketName": "",
          "AccessControlPolicy": {
              "AccessControlList": {
                  "Grant": {
                      "Grantee": {
                          "xsi:type": "CanonicalUser",
                          "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                          "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                       },
                      "Permission": "FULL_CONTROL"
                   }
              },
              "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
              "Owner": {
                  "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
              }
          },
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "acl": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
      },
      "requestID": "BD8798EACDD16751",
      "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
      "eventType": "AwsApiCall",
```

```
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "111122223333",
          "arn": "arn:aws:iam::111122223333:user/myUserName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "userName": "myUserName"
       },
      "eventTime": "2019-02-01T03:26:37Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "GetBucketVersioning",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "bucketName": "example-s3-bucket1",
          "versioning": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
    },
      "requestID": "07D681279BD94AED",
      "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
```

```
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    }
  ]
}
```

Exemple : entrées du fichier journal Amazon S3 sur Outposts

Les événements de gestion d'Amazon S3 on Outposts sont disponibles via. AWS CloudTrail Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail. En outre, vous pouvez éventuellement activer la journalisation des événements de données dans AWS CloudTrail.

Un journal de suivi est une configuration qui permet la livraison d'événements sous forme de fichiers journaux vers un compartiment S3 dans une Région que vous spécifiez. CloudTrail les journaux de vos compartiments Outposts incluent un nouveau champ`edgeDeviceDetails`, qui identifie l'Outpost où se trouve le bucket spécifié.

Les champs de journal supplémentaires incluent l'action demandée, la date et l'heure de l'action, ainsi que les paramètres de la demande. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant une PutObjectaction surs`3-outposts`.

```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "111122223333",
        "arn": "arn:aws:iam::111122223333:user/yourUserName",
        "accountId": "222222222222",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "yourUserName"
      },
      "eventTime": "2020-11-30T15:44:33Z",
      "eventSource": "s3-outposts.amazonaws.com",
      "eventName": "PutObject",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "26.29.66.20",
      "userAgent": "aws-cli/1.18.39 Python/3.4.10 Darwin/18.7.0 botocore/1.15.39",
```

```
      "requestParameters": {
        "expires": "Wed, 21 Oct 2020 07:28:00 GMT",
        "Content-Language": "english",
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
        "ObjectCannedACL": "BucketOwnerFullControl",
        "x-amz-server-side-encryption": "Aes256",
        "Content-Encoding": "gzip",
        "Content-Length": "10",
        "Cache-Control": "no-cache",
        "Content-Type": "text/html; charset=UTF-8",
        "Content-Disposition": "attachment",
        "Content-MD5": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
        "x-amz-storage-class": "Outposts",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "bucketName": "example-s3-bucket1",
        "Key": "path/upload.sh"
      },
      "responseElements": {
        "x-amz-server-side-encryption-customer-key-MD5": "wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
        "x-amz-server-side-encryption": "Aes256",
        "x-amz-version-id": "001",
        "x-amz-server-side-encryption-customer-algorithm": "Aes256",
        "ETag": "d41d8cd98f00b204e9800998ecf8427f"
      },
      "additionalEventData": {
        "CipherSuite": "ECDHE-RSA-AES128-SHA",
        "bytesTransferredIn": 10,
        "x-amz-id-2": "29xXQBV20
+xOHKItvzY1suLv1i6A52E0zOX159fpfsItYd58JhXwKxXAXI4IQkp6",
        "SignatureVersion": "SigV4",
        "bytesTransferredOut": 20,
        "AuthenticationMethod": "AuthHeader"
      },
      "requestID": "8E96D972160306FA",
      "eventID": "ee3b4e0c-ab12-459b-9998-0a5a6f2e4015",
      "readOnly": false,
      "resources": [
        {
          "accountId": "222222222222",
          "type": "AWS::S3Outposts::Object",
          "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/path/upload.sh"
```

```
        },
        {
          "accountId": "222222222222",
          "type": "AWS::S3Outposts::Bucket",
          "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/"
        }
      ],
      "eventType": "AwsApiCall",
      "managementEvent": false,
      "recipientAccountId": "444455556666",
      "sharedEventID": "02759a4c-c040-4758-b84b-7cbaaf17747a",
      "edgeDeviceDetails": {
        "type": "outposts",
        "deviceId": "op-01ac5d28a6a232904"
      },
      "eventCategory": "Data"
    }
```

Activation de la journalisation des CloudTrail événements pour les compartiments et les objets S3

Vous pouvez utiliser CloudTrail les événements de données pour obtenir des informations sur les requêtes au niveau du compartiment et de l'objet dans Amazon S3. Pour activer les événements de CloudTrail données pour tous vos compartiments ou pour une liste de compartiments spécifiques, vous devez créer un suivi manuellement dans. CloudTrail

> **Note**
>
> - Le paramètre par défaut pour CloudTrail est de rechercher uniquement les événements de gestion. Assurez-vous que vous avez activé les événements de données pour votre compte.
> - Avec un compartiment S3 qui génère une charge de travail élevée, vous pourriez rapidement générer des milliers de journaux en un temps très court. Tenez compte de la durée pendant laquelle vous choisissez d'activer CloudTrail les événements de données pour un compartiment occupé.

CloudTrail stocke les journaux d'événements de données Amazon S3 dans un compartiment S3 de votre choix. Envisagez d'utiliser un compartiment séparé Compte AWS pour mieux organiser les événements provenant de plusieurs compartiments que vous pourriez posséder dans un emplacement central afin de faciliter les requêtes et les analyses. AWS Organizations vous permet de créer un Compte AWS compte lié au compte propriétaire du bucket que vous surveillez. Pour plus d'informations, voir Qu'est-ce que c'est AWS Organizations ? dans le guide de AWS Organizations l'utilisateur.

Lorsque vous enregistrez des événements de données pour un trail in CloudTrail, vous pouvez choisir d'utiliser des sélecteurs d'événements avancés ou des sélecteurs d'événements de base. Lorsque vous créez un suivi dans la CloudTrail console à l'aide de sélecteurs d'événements avancés, dans la section Événements de données, vous pouvez sélectionner Enregistrer tous les événements pour que le modèle de sélecteur de journal enregistre tous les événements au niveau de l'objet. Lorsque vous créez un suivi dans la CloudTrail console à l'aide de sélecteurs d'événements de base, dans la section des événements de données, vous pouvez cocher la case Sélectionner tous les compartiments S3 de votre compte pour consigner tous les événements au niveau de l'objet.

> **Note**
>
> - Il est recommandé de créer une configuration de cycle de vie pour votre compartiment d'événements de données AWS CloudTrail . Configurez la configuration de cycle de vie pour supprimer périodiquement les fichiers journaux après le délai à l'issue duquel vous estimez devoir les auditer. Cela permet de réduire la quantité de données analysées par Athena pour chaque requête. Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket.
> - Pour plus d'informations sur le format de la journalisation, veuillez consulter Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.
> - Pour des exemples expliquant comment interroger les CloudTrail journaux, consultez le billet de blog AWS consacré au Big Data Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail et Amazon Athena.

Activer la journalisation des objets d'un compartiment à l'aide de la console

Vous pouvez utiliser la console Amazon S3 pour configurer un AWS CloudTrail suivi afin de consigner les événements de données relatifs aux objets d'un compartiment S3. CloudTrail prend

en charge la journalisation des opérations d'API au niveau des objets Amazon S3`GetObject`, telles que`DeleteObject`, et. `PutObject` Ces événements sont des événements de données.

Par défaut, les CloudTrail traces n'enregistrent pas les événements de données, mais vous pouvez configurer les pistes pour enregistrer les événements de données pour les compartiments S3 que vous spécifiez, ou pour enregistrer les événements de données pour tous les compartiments Amazon S3 de votre. Compte AWS Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.

CloudTrail ne renseigne pas les événements de données dans l'historique des CloudTrail événements. De plus, les actions au niveau du bucket ne sont pas toutes renseignées dans l'historique des CloudTrail événements. Pour plus d'informations sur les actions d'API au niveau du bucket Amazon S3 suivies par CloudTrail journalisation, consultez. Actions au niveau du compartiment Amazon S3 suivies par journalisation CloudTrail Pour plus d'informations sur la manière d'interroger CloudTrail les journaux, consultez l'article du centre de AWS connaissances sur l'utilisation des modèles de filtre Amazon CloudWatch Logs et d'Amazon Athena pour interroger CloudTrail les journaux.

Pour configurer un journal de suivi afin de consigner les événements de données pour un compartiment S3, vous pouvez utiliser la console AWS CloudTrail ou la console Amazon S3. Si vous configurez un suivi pour consigner les événements de données pour tous les compartiments Amazon S3 de votre choix Compte AWS, il est plus facile d'utiliser la CloudTrail console. Pour plus d'informations sur l'utilisation de la CloudTrail console pour configurer un journal des événements liés aux données S3, consultez la section Événements liés aux données dans le guide de AWS CloudTrail l'utilisateur.

> **Important**
>
> Des frais supplémentaires s'appliquent pour les événements de données. Pour en savoir plus, consultez Tarification de AWS CloudTrail.

La procédure suivante explique comment utiliser la console Amazon S3 pour configurer un journal afin de CloudTrail consigner les événements de données d'un compartiment S3.

Pour activer la journalisation des événements de CloudTrail données pour les objets d'un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment.
3. Choisissez Propriétés.
4. Sous Événements AWS CloudTrail liés aux données, sélectionnez Configurer dans CloudTrail.

   Vous pouvez créer une nouvelle CloudTrail trace ou réutiliser une trace existante et configurer les événements de données Amazon S3 pour qu'ils soient enregistrés dans votre trace. Pour plus d'informations sur la création de pistes dans la CloudTrail console, consultez la section Création et mise à jour d'une piste avec la console dans le guide de AWS CloudTrail l'utilisateur. Pour plus d'informations sur la configuration de la journalisation des événements de données Amazon S3 dans la CloudTrail console, consultez la section Journalisation des événements de données pour les objets Amazon S3 dans le guide de AWS CloudTrail l'utilisateur.

   > **Note**
   >
   > Si vous utilisez la CloudTrail console ou la console Amazon S3 pour configurer un journal afin de consigner les événements de données d'un compartiment S3, la console Amazon S3 indique que la journalisation au niveau de l'objet est activée pour le compartiment.

Pour désactiver la journalisation des événements de CloudTrail données pour les objets d'un compartiment S3

1. Connectez-vous à la CloudTrail console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudtrail/.
2. Dans le panneau de navigation de gauche, choisissez Journaux de suivi.
3. Choisissez le nom du journal de suivi que vous avez créé pour journaliser les événements de votre compartiment.
4. Sur la page de détails de votre journal de suivi, choisissez Arrêter la journalisation dans le coin supérieur droit.
5. Dans la boîte de dialogue qui s'affiche, cliquez sur Arrêter la journalisation.

Pour plus d'informations sur l'activation de la journalisation au niveau des objets lorsque vous créez un compartiment S3, consultez Créer un compartiment.

Pour plus d'informations sur la CloudTrail journalisation à l'aide de compartiments S3, consultez les rubriques suivantes :

- Affichage des propriétés d'un compartiment S3
- Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail
- Utilisation des fichiers CloudTrail journaux dans le guide de AWS CloudTrail l'utilisateur

Identification des demandes Amazon S3 à l'aide CloudTrail

Dans Amazon S3, vous pouvez identifier les demandes à l'aide d'un journal d' AWS CloudTrail événements. AWS CloudTrail est la méthode préférée pour identifier les demandes Amazon S3, mais si vous utilisez les journaux d'accès aux serveurs Amazon S3, consultezthe section called “Identification des demandes S3”.

Rubriques

- Identification des demandes adressées à Amazon S3 dans un CloudTrail journal
- Identification des demandes Amazon S3 Signature version 2 à l'aide de CloudTrail
- Identification de l'accès aux objets S3 en utilisant CloudTrail

Identification des demandes adressées à Amazon S3 dans un CloudTrail journal

Une fois que vous avez configuré CloudTrail la transmission d'événements vers un compartiment, vous devriez commencer à voir des objets se diriger vers votre compartiment de destination sur la console Amazon S3. Ils se présentent dans le format suivant :

```
s3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Les événements enregistrés par CloudTrail sont stockés sous forme d'objets gzipped JSON compressés dans votre compartiment S3. Pour rechercher efficacement les demandes, vous devez utiliser un service tel qu'Amazon Athena pour indexer et interroger les CloudTrail journaux.

Pour plus d'informations sur Athena CloudTrail et Athena, consultez la section Création de la table pour les AWS CloudTrail journaux dans Athena à l'aide de la projection de partitions dans le guide de l'utilisateur d'Amazon Athena.

Identification des demandes Amazon S3 Signature version 2 à l'aide de CloudTrail

Vous pouvez utiliser un journal d' CloudTrail événements pour identifier la version de signature d'API utilisée pour signer une demande dans Amazon S3. Cette fonctionnalité est importante étant donné que la prise en charge de Signature Version 2 va être désactivée (obsolète). Après cela, Amazon S3 n'acceptera plus les demandes utilisant Signature Version 2, et toutes les demandes devront utiliser le processus de signature Signature Version 4.

Nous vous recommandons vivement de l'utiliser CloudTrail pour déterminer si l'un de vos flux de travail utilise la signature Signature version 2. Pour y remédier, mettez à niveau vos bibliothèques et votre code afin qu'ils utilisent plutôt Signature Version 4 afin d'éviter tout impact sur votre entreprise.

Pour plus d'informations, consultez Annonce : AWS CloudTrail pour Amazon S3 ajoute de nouveaux champs pour un audit de sécurité amélioré dans AWS re:Post.

> **Note**
>
> CloudTrail les événements pour Amazon S3 incluent la version de signature dans les détails de la demande sous le nom de clé « »`additionalEventData`. Pour trouver la version de signature des demandes effectuées pour des objets dans Amazon S3`GET`, tels que,`PUT`, et des `DELETE` demandes, vous devez activer CloudTrail les événements de données. (Cette fonction est désactivée par défaut).

AWS CloudTrail est la méthode préférée pour identifier les demandes de signature version 2. Si vous utilisez les journaux d'accès au serveur Amazon S3, consultez Identification des demandes Signature Version 2 à l'aide des journaux d'accès Amazon S3.

Rubriques

- Exemples de requête Athena pour l'identification de demandes Amazon S3 Signature Version 2
- Partitionnement des données Signature Version 2

Exemples de requête Athena pour l'identification de demandes Amazon S3 Signature Version 2

Example — sélectionnez tous les événements de la version 2 de la signature et imprimez uniquement **`EventTime`**, **`S3_Action`**, **`Request_Parameters`**, **`Region`**, **`SourceIP`**, et **`UserAgent`**

Dans la requête Athena suivante, remplacez *`s3_cloudtrail_events_db.cloudtrail_table`* par vos coordonnées Athena, et augmentez ou supprimez la limite selon les besoins.

```
SELECT EventTime, EventName as S3_Action, requestParameters as Request_Parameters,
 awsregion as AWS_Region, sourceipaddress as Source_IP, useragent as User_Agent
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
LIMIT 10;
```

Example – Sélectionner tous les demandeurs qui envoient du trafic Signature Version 2

```
SELECT useridentity.arn, Count(requestid) as RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
    and json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
Group by useridentity.arn
```

Partitionnement des données Signature Version 2

Si vous devez interroger un large volume de données, vous pouvez réduire les coûts et l'exécution d'Athena en créant une table partitionnée.

Pour ce faire, créez une nouvelle table avec des partitions, comme suit.

```
  CREATE EXTERNAL TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned(
       eventversion STRING,
       userIdentity STRUCT<
           type:STRING,
           principalid:STRING,
           arn:STRING,
           accountid:STRING,
           invokedby:STRING,
           accesskeyid:STRING,
           userName:STRING,
        sessioncontext:STRUCT<
                   attributes:STRUCT<
                   mfaauthenticated:STRING,
                   creationdate:STRING>,
                   sessionIssuer:STRUCT<
```

```
                    type:STRING,
                    principalId:STRING,
                    arn:STRING,
                    accountId:STRING,
                    userName:STRING>
                >
            >,
        eventTime STRING,
        eventSource STRING,
        eventName STRING,
        awsRegion STRING,
        sourceIpAddress STRING,
        userAgent STRING,
        errorCode STRING,
        errorMessage STRING,
        requestParameters STRING,
        responseElements STRING,
        additionalEventData STRING,
        requestId STRING,
        eventId STRING,
        resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
        eventType STRING,
        apiVersion STRING,
        readOnly STRING,
        recipientAccountId STRING,
        serviceEventDetails STRING,
        sharedEventID STRING,
        vpcEndpointId STRING
    )
    PARTITIONED BY (region string, year string, month string, day string)
    ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
    STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
    LOCATION 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/';
```

Ensuite, créez les partitions individuellement. Vous ne pouvez pas obtenir des résultats à partir de dates que vous n'avez pas créées.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
```

```
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://DOC-EXAMPLE-BUCKET1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

Vous pouvez ensuite effectuer la demande en fonction de ces partitions. Vous n'avez pas besoin de charger le compartiment entier.

```
SELECT useridentity.arn,
Count(requestid) AS RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table_partitioned
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
AND region='us-east-1'
AND year='2019'
AND month='02'
AND day='19'
Group by useridentity.arn
```

Identification de l'accès aux objets S3 en utilisant CloudTrail

Vous pouvez utiliser vos journaux d' AWS CloudTrail événements pour identifier les demandes d'accès aux objets Amazon S3 relatives à des événements de données tels que `GetObject`, `DeleteObject`, et`PutObject`, et découvrir des informations supplémentaires sur ces demandes.

L'exemple suivant montre comment obtenir toutes les demandes PUT d'objets pour Amazon S3 à partir d'un journal d' AWS CloudTrail événements.

Rubriques

Exemples de requêtes Athena pour identifier les demandes d'accès aux objets Amazon S3

Dans les exemples de requêtes Athena suivants, remplacez *s3_cloudtrail_events_db.cloudtrail_table* par vos coordonnées Athena et modifiez la plage de dates si nécessaire.

Example — sélectionnez tous les événements pour lesquels il existe des demandes d'accès à des objets **PUT**, et imprimez uniquement **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** et **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'PutObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — sélectionnez tous les événements pour lesquels il existe des demandes d'accès à des objets **GET**, et imprimez uniquement **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** et **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
```

```
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'GetObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — sélectionnez tous les événements des demandeurs anonymes dans un compartiment pendant une certaine période et imprimez seulement **`EventTime`**, **`EventName`**, **`EventSource`**, **`SourceIP`**, **`UserAgent`**, **`BucketName`**, **`UserARN`** et **`AccountID`**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  userIdentity.arn as userArn,
  userIdentity.accountId
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  userIdentity.accountId = 'anonymous'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — identifiez toutes les demandes qui ont nécessité un ACL pour l'autorisation.

L'exemple de requête Amazon Athena suivant montre comment identifier toutes les requêtes de vos compartiments S3 qui ont nécessité une liste de contrôle d'accès (ACL) pour l'autorisation. Si la requête a nécessité une ACL pour l'autorisation, la valeur `aclRequired` dans `additionalEventData` est `Yes`. Si aucune ACL n'a été requise, la valeur `aclRequired` est absente. Vous pouvez utiliser ces informations pour migrer ces autorisations ACL vers les politiques de compartiment appropriées. Après avoir créé ces politiques de compartiment, vous pouvez désactiver les ACL pour ces compartiments. Pour plus d'informations sur la désactivation des ACL, consultez Conditions préalables à la désactivation des listes ACL.

```
SELECT
  eventTime,
  eventName,
  eventSource,
```

```
  sourceIpAddress,
  userAgent,
  userIdentity.arn as userArn,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  json_extract_scalar(additionalEventData, '$.aclRequired') as aclRequired
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
  AND eventTime BETWEEN '2022-05-10T00:00:00Z' and '2022-08-10T00:00:00Z'
```

> **Note**
>
> - Ces exemples de requêtes sont également utiles pour surveiller la sécurité. Vous pouvez vérifier les résultats pour les appels `PutObject` ou `GetObject` depuis des adresses IP ou des demandeurs inattendus ou non autorisés et pour identifier les demandes anonymes adressées à vos compartiments.
> - Cette requête ne récupère d'informations qu'à partir du moment où l'enregistrement a été activé.

Si vous utilisez les journaux d'accès au serveur Amazon S3, veuillez consulter Identification des demandes d’accès aux objets à l’aide des journaux d’accès Amazon S3.

Enregistrement de demandes avec journalisation des accès au serveur

La journalisation des accès au serveur fournit des enregistrements détaillés pour les demandes soumises à un compartiment. Les journaux d'accès au serveur sont utiles pour de nombreuses applications. Par exemple, les informations des journaux d'accès peuvent s'avérer utiles en cas d'audit de sécurité ou d'audit des accès. Ces informations peuvent également vous aider à en savoir plus sur votre clientèle et à comprendre votre facture Amazon S3.

> **Note**
>
> Les journaux d’accès au serveur n’enregistrent pas les informations concernant les erreurs de redirection en cas de région incorrecte pour les régions lancées après le 20 mars 2019. Les

erreurs de redirection en cas de région incorrecte se produisent lorsqu'une demande pour un objet ou un compartiment est effectuée en dehors de la région où se trouve le compartiment.

Comment activer la livraison des journaux ?

Suivez ces quelques étapes pour activer la livraison des journaux. Pour de plus amples informations, veuillez consulter Activation de la journalisation des accès au serveur Amazon S3.

1. Indiquez le nom du compartiment de destination (également appelé compartiment cible). Amazon S3 enregistrera dans ce compartiment les journaux d'accès en tant qu'objets. Les compartiments source et de destination doivent être dans la même Région AWS et appartenir au même compte. Le compartiment de destination ne doit pas avoir de configuration de période de rétention par défaut pour le verrouillage d'objet S3. Le paiement par le demandeur ne doit pas non plus être activé pour le compartiment de destination.

   Les journaux peuvent être fournis dans n'importe quel compartiment que vous possédez qui est situé dans la même Région que le compartiment source, y compris le compartiment source lui-même. Mais pour simplifier la gestion des journaux, nous vous recommandons d'enregistrer les journaux d'accès dans un autre compartiment.

   Lorsque votre compartiment source et votre compartiment de destination correspondent au même compartiment, des journaux supplémentaires sont créés pour les journaux qui sont écrits dans le compartiment, ce qui crée une boucle infinie de journaux. Nous ne recommandons pas de procéder ainsi, car cela peut entraîner une légère augmentation de votre facture de stockage. En outre, les journaux supplémentaires concernant les journaux peuvent rendre plus difficile la recherche du journal que vous recherchez.

   Si vous choisissez d'enregistrer les journaux d'accès dans le compartiment source, nous vous recommandons de spécifier un préfixe de destination (également appelé préfixe cible) pour toutes les clés d'objets journaux. Lorsque vous spécifiez un préfixe, tous les noms des objets journaux commencent par une chaîne commune, ce qui facilite l'identification des objets journaux.
2. (Facultatif) Affectez un préfixe de destination à toutes les clés d'objets journaux Amazon S3. Le préfixe de destination (également appelé préfixe cible) vous aide à localiser les objets journaux. Par exemple, si vous spécifiez la valeur de préfixe `logs/`, chaque objet journal créé par Amazon S3 commence par le préfixe `logs/` dans sa clé, par exemple :

```
logs/2013-11-01-21-32-16-E568B2907131C0C0
```

Si vous spécifiez la valeur de préfixe `logs`, l'objet journal apparaît comme suit :

```
logs2013-11-01-21-32-16-E568B2907131C0C0
```

Les préfixes sont également utiles pour distinguer les compartiments sources lorsque plusieurs compartiments sont journalisés dans le même compartiment de destination.

Ce préfixe peut également s'avérer utile lors de la suppression des journaux. Par exemple, vous pouvez définir une règle de configuration du cycle de vie pour qu'Amazon S3 supprime les objets dotés d'un préfixe spécifique. Pour plus d'informations, consultez Suppression des fichiers journaux Amazon S3.

3. (Facultatif) Définissez des autorisations pour que d'autres utilisateurs puissent accéder aux journaux générés. Par défaut, seul le propriétaire du compartiment possède un accès total aux objets des journaux. Si votre compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets S3 afin de désactiver les listes de contrôle d'accès (ACL), vous ne pouvez pas accorder d'autorisations dans des octrois de destination (également appelés octrois cibles) qui utilisent des listes ACL. Toutefois, vous pouvez mettre à jour votre politique de compartiment pour le compartiment de destination afin d'accorder l'accès à d'autres personnes. Pour plus d'informations, consultez Identity and Access Management pour Amazon S3 et Autorisations de diffusion de journaux.
4. (Facultatif) Définissez un format de clé d'objet journal pour les fichiers journaux. Deux options s'offrent à vous pour le format de clé d'objet journal (également appelé format de clé d'objet cible) :
   - N on-date-based partitionnement : il s'agit du format de clé de l'objet journal d'origine. Si vous choisissez ce format, le format de clé de fichier journal apparaît comme suit :

     ```
     [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
     ```

     Par exemple, si vous spécifiez `logs/` comme préfixe, vos objets journaux sont nommés comme suit :

     ```
     logs/2013-11-01-21-32-16-E568B2907131C0C0
     ```

- Partitionnement basé sur la date : si vous choisissez un partitionnement basé sur la date, vous pouvez choisir l'heure de l'événement ou l'heure de livraison du fichier journal comme source de date utilisée dans le format de journal. Ce format facilite l'interrogation des journaux.

  Si vous choisissez un partitionnement basé sur la date, le format de clé de fichier journal apparaît comme suit :

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/
  [DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

  Par exemple, si vous spécifiez `logs/` comme préfixe cible, vos objets journaux sont nommés comme suit :

  ```
  logs/123456789012/us-west-2/DOC-EXAMPLE-SOURCE-
  BUCKET/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
  ```

  Pour la livraison de l'heure de livraison, l'heure indiquée dans les noms des fichiers journaux correspond à l'heure de livraison des fichiers journaux.

  Pour la livraison de l'heure d'événement, l'année, le mois et le jour correspondent au jour où l'événement s'est produit, et l'heure, les minutes et les secondes sont définies sur `00` dans la clé. Les journaux livrés dans ces fichiers journaux se rapportent à un jour spécifique uniquement.

Si vous configurez vos journaux via AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST Amazon S3, utilisez-le `TargetObjectKeyFormat` pour spécifier le format de clé de l'objet du journal. Pour spécifier le non-date-based partitionnement, utilisez`SimplePrefix`. Pour spécifier un partitionnement basé sur la date, utilisez `PartitionedPrefix`. Si vous utilisez `PartitionedPrefix`, utilisez `PartitionDateSource` pour spécifier `EventTime` ou `DeliveryTime`.

Pour `SimplePrefix`, le format de clé de fichier journal apparaît comme suit :

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Pour `PartitionedPrefix` avec l'heure d'événement ou l'heure de livraison, le format de clé de fichier journal apparaît comme suit :

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Format de clé d'objet journal

Amazon S3 utilise les formats de clé d’objet suivants pour les objets journaux qu’il charge dans le compartiment de destination :

- N on-date-based partitionnement : il s'agit du format de clé de l'objet journal d'origine. Si vous choisissez ce format, le format de clé de fichier journal apparaît comme suit :

  ```
  [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

- Partitionnement basé sur la date : si vous choisissez un partitionnement basé sur la date, vous pouvez choisir l’heure de l’événement ou l’heure de livraison du fichier journal comme source de date utilisée dans le format de journal. Ce format facilite l’interrogation des journaux.

  Si vous choisissez un partitionnement basé sur la date, le format de clé de fichier journal apparaît comme suit :

  ```
  [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
  [YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
  ```

Dans la clé d’objet journal, `YYYY`, `MM`, `DD`, `hh`, `mm` et `ss` correspondent (respectivement) à l’année, au mois, au jour, aux heures, aux minutes et aux secondes. Ces dates et heures sont exprimées en heure UTC (temps universel coordonné).

Un fichier journal distribué à un moment précis peut contenir des enregistrements écrits à tout moment avant ce moment. Il n'existe aucun moyen de savoir si tous les enregistrements d'un intervalle de temps donné ont été distribués.

Le composant `UniqueString` de la clé empêche le remplacement des fichiers. Il n'a aucune signification, et doit être ignoré par les logiciels de traitement des journaux.

Comment sont distribués les journaux ?

Amazon S3 collecte régulièrement les enregistrements des journaux d'accès, consolide ces enregistrements dans des fichiers journaux, puis charge les fichiers journaux dans le compartiment de destination comme objets journaux. Si vous activez la journalisation sur plusieurs compartiments sources qui identifient le même compartiment de destination, ce dernier dispose de journaux d'accès pour tous ces compartiments sources. Cependant, chaque objet journal contient des enregistrements de journal d'accès pour un compartiment source spécifique.

Amazon S3 utilise un compte de livraison de journaux spécial pour écrire des journaux d'accès au serveur. Ces journaux sont sujets aux restrictions habituelles de contrôle d'accès en écriture. Nous vous recommandons de mettre à jour la politique de compartiment sur le compartiment de destination pour accorder l'accès au principal du service de journalisation (`logging.s3.amazonaws.com`) pour la livraison des journaux des accès. Vous pouvez également accorder l'accès pour la livraison des journaux des d'accès au groupe de livraison des journaux S3 via votre liste de contrôle d'accès (ACL) de compartiment. Toutefois, il n'est pas recommandé d'accorder l'accès au groupe de livraison des journaux S3 en utilisant votre liste ACL de compartiment.

Lorsque vous activez la journalisation des accès au serveur et que vous accordez l'accès à la livraison des journaux des accès via votre politique de compartiment de destination, vous devez mettre à jour la politique pour autoriser l'accès `s3:PutObject` pour le principal du service de journalisation. Si vous utilisez la console Amazon S3 pour activer la journalisation des accès au serveur, la console met automatiquement à jour la politique de compartiment de destination, afin d'accorder ces autorisations au principal du service de journalisation. Pour plus d'informations sur l'accord d'autorisations pour la livraison de journaux d'accès au serveur, consultez Autorisations de diffusion de journaux.

> **Note**
>
> S3 ne prend pas en charge la livraison de CloudTrail journaux ou de journaux d'accès au serveur au demandeur ou au propriétaire du compartiment pour les demandes de point de terminaison VPC lorsque la politique de point de terminaison du VPC les refuse ou pour les demandes qui échouent avant que la politique VPC ne soit évaluée.

Paramètre bucket owner enforced (propriétaire du compartiment imposé) pour S3 Object Ownership (Propriété de l'objet S3)

Si le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, les listes ACL sont désactivées et n'affectent plus les autorisations. Vous devez mettre à jour la politique de compartiment sur le compartiment de destination pour accorder l'accès au principal du service de journalisation. Pour en savoir plus sur la propriété des objets, veuillez consulter Octroi de l'accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur.

Livraison des journaux du serveur dans la mesure du possible

Les enregistrements des journaux d'accès au serveur sont fournis dans la mesure du possible. La plupart des demandes pour un compartiment correctement configuré pour l'enregistrement se traduisent par un enregistrement de journal distribué. La plupart des enregistrements de journal sont fournis quelques heures après leur enregistrement, mais ils peuvent être livrés plus fréquemment.

L'exhaustivité et la disponibilité de la journalisation du serveur ne sont pas garanties. Il se peut que l'enregistrement du journal pour une demande particulière soit fourni bien après le traitement de la demande, ou ne soit fourni du tout. Il est même possible que vous voyiez une duplication d'un enregistrement de journal. Le but des journaux du serveur est de vous donner une idée de la nature du trafic dans le compartiment. Bien qu'il soit rare de perdre des enregistrements de journaux ou de constater une duplication d'enregistrements de journaux, sachez que la journalisation du serveur n'a pas pour but de comptabiliser toutes les demandes.

En raison de la nature de la journalisation du serveur, visant un effort maximal, vos rapports d'utilisation peuvent inclure une ou plusieurs demandes qui n'apparaissent pas dans un journal de serveur livré. Ces rapports d'utilisation se trouvent sous Rapports d'utilisation et de coût dans la console AWS Billing and Cost Management .

Les changements de statut de la journalisation des compartiments prennent effet au fil du temps

Les modifications du statut de l'état de journalisation d'un compartiment prennent du temps avant d'affecter réellement la distribution des fichiers journaux. Par exemple, si vous activez la journalisation pour un compartiment, certaines demandes faites dans l'heure qui suit peuvent être journalisées, alors que d'autres ne le sont pas. Supposons que vous changiez le compartiment de destination de la journalisation du compartiment A au compartiment B. L'heure suivante, certains journaux peuvent continuer à être livrés dans le compartiment A, tandis que d'autres peuvent être livrés dans le nouveau compartiment de destination B. Dans tous les cas, les nouveaux paramètres finissent par prendre effet sans aucune autre action de votre part.

Pour plus d'informations sur la journalisation et les fichiers journaux, consultez les sections suivantes :

Rubriques

Activation de la journalisation des accès au serveur Amazon S3

La journalisation des accès au serveur fournit des enregistrements détaillés des demandes soumises à un compartiment Amazon S3. Les journaux d'accès au serveur sont utiles pour de nombreuses applications. Par exemple, les informations des journaux d'accès peuvent s'avérer utiles en cas d'audit de sécurité ou d'audit des accès. Ces informations peuvent également vous aider à en savoir plus sur votre clientèle et à comprendre votre facture Amazon S3.

Par défaut, Amazon S3 ne collecte pas les journaux d'accès au serveur. Lorsque vous activez la journalisation, Amazon S3 fournit les journaux d’accès pour un compartiment source dans un compartiment de destination (également appelé compartiment cible) de votre choix. Le compartiment de destination doit se trouver dans la même Région AWS et dans le même Compte AWS que le compartiment source.

Un enregistrement de journal d'accès contient des détails relatifs aux demandes soumises à un compartiment. Ces informations peuvent comprendre le type de demande, les ressources spécifiées dans la demande, ainsi que l'heure et la date du traitement de la demande. Pour plus d’informations sur les principes de base de la journalisation, consultez Enregistrement de demandes avec journalisation des accès au serveur.

> **Important**
>
> - L'activation de la journalisation des accès au serveur sur un compartiment Amazon S3 n'entraîne aucuns frais supplémentaires. Toutefois, les fichiers journaux qui vous sont fournis par le système augmentent les coûts de stockage habituels. Notez que vous pouvez supprimer les fichiers journaux à tout moment. Nous n’évaluons pas les frais de transfert de données pour la livraison des fichiers journaux, mais nous facturons les frais standard de transfert de données pour l’accès aux fichiers journaux.

- La journalisation des accès au serveur de votre compartiment de destination ne doit pas être activée. Les journaux peuvent être fournis dans n'importe quel compartiment que vous possédez qui est situé dans la même Région que le compartiment source, y compris le compartiment source lui-même. Toutefois, la livraison de journaux vers le compartiment source entraîne une boucle infinie de journaux et n'est pas recommandée. Pour simplifier la gestion des journaux, nous vous recommandons d'enregistrer les journaux d'accès dans un autre compartiment. Pour plus d'informations, consultez Comment activer la livraison des journaux ?.
- Les compartiments S3 sur lesquels le verrouillage d'objet S3 est activé ne peuvent pas être utilisés comme compartiments de destination pour les journaux d'accès au serveur. Votre compartiment de destination ne doit pas avoir de configuration de période de rétention par défaut.
- Le paiement par le demandeur ne doit pas être activé pour le compartiment de destination.
- Vous pouvez utiliser le chiffrement de compartiment par défaut sur le compartiment de destination seulement si vous utilisez le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3), qui utilise la norme Advanced Encryption Standard à 256 bits (AES-256). Le chiffrement côté serveur par défaut avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) n'est pas pris en charge.

Vous pouvez activer ou désactiver la journalisation des accès au serveur à l'aide de la console Amazon S3, de l'API Amazon S3, de l' AWS Command Line Interface (AWS CLI) ou de kits SDK AWS .

Autorisations de diffusion de journaux

Amazon S3 utilise un compte de livraison de journaux spécial pour écrire des journaux d'accès au serveur. Ces journaux sont sujets aux restrictions habituelles de contrôle d'accès en écriture. Pour la livraison des journaux des accès, vous devez accorder au principal du service de journalisation (`logging.s3.amazonaws.com`) l'accès à votre compartiment de destination.

Pour accorder des autorisations à Amazon S3 pour la livraison des journaux, vous pouvez utiliser une politique de compartiment ou des listes de contrôle d'accès (ACL) de compartiment, en fonction des paramètres de propriété d'objets S3 de votre compartiment de destination. Toutefois, nous vous recommandons d'utiliser une politique de compartiment plutôt que des listes ACL.

Paramètre bucket owner enforced (propriétaire du compartiment imposé) pour S3 Object Ownership (Propriété de l'objet S3)

Si le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, les listes ACL sont désactivées et n'affectent plus les autorisations. Dans ce cas, vous devez mettre à jour la politique de compartiment pour le compartiment de destination afin d'accorder l'accès au principal du service de journalisation. Vous ne pouvez pas mettre à jour la liste ACL de votre compartiment pour accorder l'accès au groupe de mise à disposition des journaux S3. Vous ne pouvez pas non plus inclure des octrois de destination (également appelés octrois cibles) dans votre configuration PutBucketLogging.

Pour plus d'informations sur la migration des listes ACL de compartiment existantes pour la livraison du journal d'accès vers une stratégie de compartiment, consultez Octroi de l'accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur. Pour en savoir plus sur la propriété des objets, veuillez consulter Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment. Lorsque vous créez des compartiments, les listes ACL sont désactivées par défaut.

Octroi d'accès à l'aide d'une politique de compartiment

Pour accorder l'accès en utilisant la politique de compartiment sur le compartiment de destination, mettez à jour la politique de compartiment afin d'accorder l'autorisation `s3:PutObject` au principal du service de journalisation. Si vous utilisez la console Amazon S3 pour activer la journalisation des accès au serveur, la console met automatiquement à jour la politique de compartiment sur le compartiment de destination, afin d'accorder cette autorisation au principal du service de journalisation. Si vous activez la journalisation des accès au serveur par programmation, vous devez mettre à jour manuellement la politique de compartiment pour le compartiment de destination afin d'accorder l'accès au principal du service de journalisation.

Pour un exemple de politique de compartiment qui accorde l'accès au principal du service de journalisation, consultez the section called "Octroi d'autorisations au principal du service de journalisation à l'aide d'une politique de compartiment".

Octroi d'accès à l'aide de listes ACL de compartiment

Vous pouvez également utiliser les listes ACL de compartiment pour accorder l'accès aux journaux d'accès. Vous ajoutez une entrée d'accord à la liste ACL du compartiment qui accorde les autorisations `WRITE` et `READ_ACP` sur le groupe de mise à disposition des journaux S3. Toutefois, il n'est pas recommandé d'accorder l'accès au groupe de livraison des journaux S3 en utilisant des

listes ACL de compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment. Pour plus d'informations sur la migration des listes ACL de compartiment existantes pour la livraison du journal d'accès vers une stratégie de compartiment, consultez Octroi de l'accès au groupe de livraison des journaux S3 pour la journalisation des accès au serveur. Pour obtenir un exemple de liste ACL qui accorde l'accès au principal du service de journalisation, consultez the section called "Octroi d'autorisations au groupe de livraison des journaux à l'aide d'une liste ACL de compartiment".

Octroi d'autorisations au principal du service de journalisation à l'aide d'une politique de compartiment

Cet exemple de politique de compartiment accorde l'autorisation `s3:PutObject` au principal du service de journalisation (`logging.s3.amazonaws.com`). Pour utiliser cette politique de compartiment, remplacez *`user input placeholders`* par vos propres informations. Dans la politique suivante, *`example-s3-destination-bucket`* il s'agit du compartiment de destination dans lequel les journaux d'accès au serveur seront fournis et *`example-s3-source-bucket`* du compartiment source. *`EXAMPLE-LOGGING-PREFIX`*est le préfixe de destination facultatif (également appelé préfixe cible) que vous souhaitez utiliser pour vos objets de journal. *`SOURCE-ACCOUNT-ID`*est celui Compte AWS qui possède le compartiment source.

> **Note**
>
> Si votre politique de compartiment contient des instructions `Deny`, veillez à ce qu'elles n'empêchent pas Amazon S3 de livrer les journaux d'accès.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::example-s3-destination-bucket/EXAMPLE-LOGGING-
PREFIX*",
            "Condition": {
```

```
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::example-s3-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

Octroi d'autorisations au groupe de livraison des journaux à l'aide d'une liste ACL de compartiment

> **Note**
>
> Pour des raisons de sécurité, Amazon S3 désactive les listes de contrôle d'accès (ACL) par défaut dans tous les nouveaux compartiments. Pour plus d'informations sur les autorisations ACL dans la console Amazon S3, consultez Configuration des listes ACL.

Nous ne recommandons pas cette approche, mais vous pouvez accorder des autorisations au groupe de livraison des journaux en utilisant une liste ACL de compartiment. Toutefois, si le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriétés d'objets, vous ne pouvez pas définir de listes ACL de compartiment ni d'objet. Vous ne pouvez pas non plus inclure des octrois de destination (également appelés octrois cibles) dans votre configuration PutBucketLogging. À la place, vous devez utiliser une politique de compartiment pour accorder l'accès au principal du service de journalisation (`logging.s3.amazonaws.com`). Pour plus d'informations, consultez Autorisations de diffusion de journaux.

Dans la liste ACL de compartiment, le groupe de livraison des journaux est représenté par l'URL suivante :

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Pour accorder des autorisations `WRITE` et `READ_ACP` (lecture de liste ACL), ajoutez les octrois suivants à la liste ACL du compartiment de destination :

```
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
```

```
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>WRITE</Permission>
</Grant>
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>READ_ACP</Permission>
</Grant>
```

Pour obtenir des exemples d'ajout d'attributions ACL par programmation, veuillez consulter Configuration des listes ACL.

> **Important**
>
> Lorsque vous activez la journalisation de l'accès au serveur Amazon S3 AWS CloudFormation à l'aide d'un compartiment et que vous utilisez des ACL pour accorder l'accès au groupe de mise à disposition des journaux S3, vous devez également ajouter « » `AccessControl": "LogDeliveryWrite"` à votre CloudFormation modèle. Cela est important car vous ne pouvez accorder ces autorisations qu'en créant une ACL pour le bucket, mais vous ne pouvez pas créer de ACL personnalisées pour les buckets qu'il contient. CloudFormation Vous ne pouvez utiliser que des ACL prédéfinies avec CloudFormation.

Pour activer la journalisation des accès au serveur

Pour activer la journalisation des accès au serveur à l'aide de la console Amazon S3, de l'API REST Amazon S3, des kits de AWS développement logiciel (SDK) AWS CLI, suivez les procédures suivantes.

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer la journalisation des accès au serveur.
3. Choisissez Propriétés.

4. Dans la section Server access logging (Journalisation des accès au serveur) choisissez Edit (Modifier).
5. Sous Consignation des accès au serveur, choisissez Activer.
6. Sous Compartiment de destination, spécifiez un compartiment et un préfixe facultatif. Si vous spécifiez un préfixe, nous vous recommandons d'inclure une barre oblique (/) après le préfixe pour faciliter la recherche des journaux.

> **Note**
>
> La spécification d'un préfixe avec une barre oblique (/) simplifie la localisation des objets journaux. Par exemple, si vous spécifiez la valeur de préfixe `logs/`, chaque objet journal créé par Amazon S3 commence par le préfixe `logs/` dans sa clé, comme suit :
>
> ```
> logs/2013-11-01-21-32-16-E568B2907131C0C0
> ```
>
> Si vous spécifiez la valeur de préfixe `logs`, l'objet journal apparaît comme suit :
>
> ```
> logs2013-11-01-21-32-16-E568B2907131C0C0
> ```

7. Sous Format de clé d'objet journal, effectuez l'une des opérations suivantes :
   - Pour choisir le non-date-based partitionnement, choisissez [DestinationPrefix] [YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - []. UniqueString
   - Pour choisir le partitionnement basé sur la date, choisissez [DestinationPrefix] [SourceAccountId]/[SourceRegion]/[YYYYSourceBucket]/[MM]/[DD]/[YYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString], puis choisissez l'heure de l'événement S3 ou l'heure de livraison du fichier journal.
8. Sélectionnez Enregistrer les modifications.

Lorsque vous activez la journalisation des accès au serveur sur un compartiment, la console active la journalisation sur le compartiment source et met à jour la politique de compartiment pour le compartiment de destination afin d'accorder l'autorisation `s3:PutObject` au principal du service de journalisation (`logging.s3.amazonaws.com`). Pour plus d'informations sur cette stratégie de compartiment, consultez Octroi d'autorisations au principal du service de journalisation à l'aide d'une politique de compartiment.

Vous pouvez afficher les journaux dans le compartiment de destination. Une fois que vous avez activé la journalisation des accès au serveur, cela peut prendre quelques heures avant que les journaux sont livrés dans le compartiment cible. Pour plus d'informations sur la façon et le moment de livraison des journaux, consultez Comment sont distribués les journaux ?.

Pour plus d'informations, consultez Affichage des propriétés d'un compartiment S3.

Utilisation de l'API REST

Pour activer la journalisation, vous envoyez une demande PutBucketLogging pour ajouter la configuration de journalisation sur le compartiment source. La demande spécifie le compartiment de destination (également appelé compartiment cible) et, si vous le souhaitez, le préfixe à utiliser avec toutes les clés d'objets journaux.

L'exemple suivant identifie *example-s3-destination-bucket* comme compartiment de destination et *logs/* comme préfixe.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>example-s3-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

L'exemple suivant identifie *example-s3-destination-bucket* comme compartiment de destination, *logs/* comme préfixe et `EventTime` comme format de clé d'objet journal.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>example-s3-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
         <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
  </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Les objets des journaux sont écrits et détenus par le compte de livraison des journaux S3, et le propriétaire du compartiment possède les autorisations complètes sur ces objets. Vous pouvez éventuellement utiliser des octrois de destination (également appelés octrois cibles) pour accorder des autorisations aux autres utilisateurs, afin qu'ils puissent accéder aux journaux. Pour plus d'informations, consultez PutBucketLogging.

> **Note**
>
> Si le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, vous ne pouvez pas utiliser les octrois de destination pour accorder des autorisations à d'autres utilisateurs. Pour accorder des autorisations à d'autres, vous pouvez mettre à jour la politique de compartiment sur le compartiment de destination. Pour plus d'informations, consultez Autorisations de diffusion de journaux.

Pour récupérer la configuration de journalisation sur un compartiment, utilisez l'opération d'API GetBucketLogging.

Pour supprimer la configuration de la journalisation, vous envoyez une demande `PutBucketLogging` avec un paramètre `BucketLoggingStatus` vide :

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Pour activer la connexion à un compartiment, vous pouvez utiliser l'API Amazon S3 ou les bibliothèques d'encapsulation du AWS SDK.

Utilisation des AWS SDK

Les exemples suivants activent la journalisation sur un compartiment. Vous devez créer deux compartiments : un compartiment source et un compartiment de destination (cible). Les exemples commencent par mettre à jour la liste ACL de compartiment sur le compartiment de destination. Ils accordent alors au groupe de livraison des journaux les autorisations nécessaires pour écrire des journaux dans le compartiment de destination, puis activent la journalisation sur le compartiment source.

Ces exemples ne fonctionnent pas sur les compartiments de destination qui utilisent le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets.

Si le compartiment de destination (cible) utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, vous ne pouvez pas définir les listes ACL de compartiment ni d'objet. Vous ne pouvez pas non plus inclure d'autorisations de destination (cible) dans votre PutBucketLoggingconfiguration. Vous devez utiliser une stratégie de compartiment pour accorder l'accès au principal du service de journalisation (`logging.s3.amazonaws.com`). Pour plus d'informations, consultez Autorisations de diffusion de journaux.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
```

```
            string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
            string accountId = _configuration["AccountId"];

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
            IAmazonS3 client = new AmazonS3Client();

            try
            {
                // Update bucket policy for target bucket to allow delivery of
logs to it.
                await SetBucketPolicyToAllowLogDelivery(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix,
                    accountId);

                // Enable logging on the source bucket.
                await EnableLoggingAsync(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error: {e.Message}");
            }
        }

        /// <summary>
        /// This method grants appropriate permissions for logging to the
        /// Amazon S3 bucket where the logs will be stored.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
used
        /// to apply the bucket policy.</param>
        /// <param name="sourceBucketName">The name of the source bucket.</param>
        /// <param name="logBucketName">The name of the bucket where logging
        /// information will be stored.</param>
```

```
        /// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
        /// <param name="accountId">The account id of the account where the
source bucket exists.</param>
        /// <returns>Async task.</returns>
        public static async Task SetBucketPolicyToAllowLogDelivery(
            IAmazonS3 client,
            string sourceBucketName,
            string logBucketName,
            string logPrefix,
            string accountId)
        {
            var resourceArn = @"""arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @"*""";

            var newPolicy = @"{
                                ""Statement"":[{
                                ""Sid"": ""S3ServerAccessLogsPolicy"",
                                ""Effect"": ""Allow"",
                                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                                ""Action"": [""s3:PutObject""],
                                ""Resource"": [" + resourceArn + @"],
                                ""Condition"": {
                                ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @""" },
                                ""StringEquals"": { ""aws:SourceAccount"": """ +
accountId + @""" }
                                        }
                                    }]
                                }";
            Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
            Console.WriteLine(newPolicy);

            PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
            {
                BucketName = logBucketName,
                Policy = newPolicy,
            };
            await client.PutBucketPolicyAsync(putRequest);
            Console.WriteLine("Policy applied.");
        }
```

```
        /// <summary>
        /// This method enables logging for an Amazon S3 bucket. Logs will be
 stored
        /// in the bucket you selected for logging. Selected prefix
        /// will be prepended to each log object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
 used
        /// to configure and apply logging to the selected Amazon S3 bucket.</
param>
        /// <param name="bucketName">The name of the Amazon S3 bucket for which
 you
        /// wish to enable logging.</param>
        /// <param name="logBucketName">The name of the Amazon S3 bucket where
 logging
        /// information will be stored.</param>
        /// <param name="logObjectKeyPrefix">The prefix to prepend to each
        /// object key.</param>
        /// <returns>Async task.</returns>
        public static async Task EnableLoggingAsync(
            IAmazonS3 client,
            string bucketName,
            string logBucketName,
            string logObjectKeyPrefix)
        {
            Console.WriteLine($"Enabling logging for bucket {bucketName}.");
            var loggingConfig = new S3BucketLoggingConfig
            {
                TargetBucketName = logBucketName,
                TargetPrefix = logObjectKeyPrefix,
            };

            var putBucketLoggingRequest = new PutBucketLoggingRequest
            {
                BucketName = bucketName,
                LoggingConfig = loggingConfig,
            };
            await client.PutBucketLoggingAsync(putBucketLoggingRequest);
            Console.WriteLine($"Logging enabled.");
        }

        /// <summary>
        /// Loads configuration from settings files.
        /// </summary>
```

```
        public static void LoadConfig()
        {
            _configuration = new ConfigurationBuilder()
                .SetBasePath(Directory.GetCurrentDirectory())
                .AddJsonFile("settings.json") // Load settings from .json file.
                .AddJsonFile("settings.local.json", true) // Optionally, load
  local settings.
                .Build();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketLoggingà la section Référence des AWS SDK for .NET API.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLoggingStatus;
import software.amazon.awssdk.services.s3.model.LoggingEnabled;
import software.amazon.awssdk.services.s3.model.PartitionedPrefix;
import software.amazon.awssdk.services.s3.model.PutBucketLoggingRequest;
import software.amazon.awssdk.services.s3.model.TargetObjectKeyFormat;

// Class to set a bucket policy on a target S3 bucket and enable server access
 logging on a source S3 bucket.
public class ServerAccessLogging {
    private static S3Client s3Client;

    public static void main(String[] args) {
        String sourceBucketName = "SOURCE-BUCKET";
        String targetBucketName = "TARGET-BUCKET";
        String sourceAccountId = "123456789012";
        String targetPrefix = "logs/";

        // Create S3 Client.
        s3Client = S3Client.builder().
                region(Region.US_EAST_2)
                .build();
```

```
        // Set a bucket policy on the target S3 bucket to enable server access
 logging by granting the
        // logging.s3.amazonaws.com principal permission to use the PutObject
 operation.
        ServerAccessLogging serverAccessLogging = new ServerAccessLogging();
        serverAccessLogging.setTargetBucketPolicy(sourceAccountId, sourceBucketName,
 targetBucketName);

        // Enable server access logging on the source S3 bucket.
        serverAccessLogging.enableServerAccessLogging(sourceBucketName,
 targetBucketName,
                targetPrefix);

    }

    // Function to set a bucket policy on the target S3 bucket to enable server
 access logging by granting the
    // logging.s3.amazonaws.com principal permission to use the PutObject operation.
    public void setTargetBucketPolicy(String sourceAccountId, String
 sourceBucketName, String targetBucketName) {
        String policy = "{\n" +
                "    \"Version\": \"2012-10-17\",\n" +
                "    \"Statement\": [\n" +
                "        {\n" +
                "            \"Sid\": \"S3ServerAccessLogsPolicy\",\n" +
                "            \"Effect\": \"Allow\",\n" +
                "            \"Principal\": {\"Service\": \"logging.s3.amazonaws.com
\"},\n" +
                "            \"Action\": [\n" +
                "                \"s3:PutObject\"\n" +
                "            ],\n" +
                "            \"Resource\": \"arn:aws:s3:::" + targetBucketName + "/*
\",\n" +
                "            \"Condition\": {\n" +
                "                \"ArnLike\": {\n" +
                "                    \"aws:SourceArn\": \"arn:aws:s3:::" +
 sourceBucketName + "\"\n" +
                "                },\n" +
                "                \"StringEquals\": {\n" +
                "                    \"aws:SourceAccount\": \"" + sourceAccountId +
 "\"\n" +
                "                }\n" +
                "            }\n" +
                "        }\n" +
```

```
                "    ]\n" +
                "}";
        s3Client.putBucketPolicy(b -> b.bucket(targetBucketName).policy(policy));
    }

    // Function to enable server access logging on the source S3 bucket.
    public void enableServerAccessLogging(String sourceBucketName, String
 targetBucketName,
            String targetPrefix) {
        TargetObjectKeyFormat targetObjectKeyFormat =
 TargetObjectKeyFormat.builder()

 .partitionedPrefix(PartitionedPrefix.builder().partitionDateSource("EventTime").build())
                .build();
        LoggingEnabled loggingEnabled = LoggingEnabled.builder()
                .targetBucket(targetBucketName)
                .targetPrefix(targetPrefix)
                .targetObjectKeyFormat(targetObjectKeyFormat)
                .build();
        BucketLoggingStatus bucketLoggingStatus = BucketLoggingStatus.builder()
                .loggingEnabled(loggingEnabled)
                .build();
        s3Client.putBucketLogging(PutBucketLoggingRequest.builder()
                .bucket(sourceBucketName)
                .bucketLoggingStatus(bucketLoggingStatus)
                .build());
    }

}
```

En utilisant le AWS CLI

Nous vous recommandons de créer un compartiment de journalisation dédié dans chaque compartiment dans Région AWS lequel vous avez des compartiments S3. Ensuite, faites en sorte que les journaux d’accès Amazon S3 soient livrés dans ce compartiment S3. Pour plus d'informations et des exemples, consultez put-bucket-logging dans la référence AWS CLI .

Si le compartiment de destination (cible) utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d’objets, vous ne pouvez pas définir les listes ACL de compartiment ni d’objet. Vous ne pouvez pas non plus inclure d'autorisations de destination (cible) dans votre PutBucketLoggingconfiguration. Vous devez utiliser une stratégie de compartiment pour accorder

l'accès au principal du service de journalisation (`logging.s3.amazonaws.com`). Pour plus d'informations, consultez Autorisations de diffusion de journaux.

Example – Activer des journaux d'accès avec cinq compartiments répartis dans deux Régions

Dans cet exemple, vous disposez des cinq compartiments suivants :

- `1-DOC-EXAMPLE-BUCKET1-us-east-1`
- `2-DOC-EXAMPLE-BUCKET1-us-east-1`
- `3-DOC-EXAMPLE-BUCKET1-us-east-1`
- `1-DOC-EXAMPLE-BUCKET1-us-west-2`
- `2-DOC-EXAMPLE-BUCKET1-us-west-2`

> **Note**
>
> La dernière étape de la procédure suivante fournit des exemples de scripts bash que vous pouvez utiliser pour créer vos compartiments de journalisation et activer la journalisation des accès au serveur sur ces compartiments. Pour utiliser ces scripts, vous devez créer les fichiers `policy.json` et `logging.json`, comme décrit dans la procédure suivante.

1. Créez deux compartiments de destination de journalisation dans les régions USA Ouest (Oregon) et USA Est (Virginie du Nord) et donnez-leur les noms suivants :
   - `DOC-EXAMPLE-BUCKET1-logs-us-east-1`
   - `DOC-EXAMPLE-BUCKET1-logs-us-west-2`
2. Plus tard dans ces étapes, vous activerez la journalisation des accès au serveur comme suit :
   - `1-DOC-EXAMPLE-BUCKET1-us-east-1` consigne dans le compartiment S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` avec le préfixe `1-DOC-EXAMPLE-BUCKET1-us-east-1`.
   - `2-DOC-EXAMPLE-BUCKET1-us-east-1` consigne dans le compartiment S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` avec le préfixe `2-DOC-EXAMPLE-BUCKET1-us-east-1`.
   - `3-DOC-EXAMPLE-BUCKET1-us-east-1` consigne dans le compartiment S3 `DOC-EXAMPLE-BUCKET1-logs-us-east-1` avec le préfixe `3-DOC-EXAMPLE-BUCKET1-us-east-1`.
   - `1-DOC-EXAMPLE-BUCKET1-us-west-2` consigne dans le compartiment S3 `DOC-EXAMPLE-BUCKET1-logs-us-west-2` avec le préfixe `1-DOC-EXAMPLE-BUCKET1-us-west-2`.

- `2-DOC-EXAMPLE-BUCKET1-us-west-2` consigne dans le compartiment S3 `DOC-EXAMPLE-BUCKET1-logs-us-west-2` avec le préfixe `2-DOC-EXAMPLE-BUCKET1-us-west-2`.

3. Pour chaque compartiment de journalisation de destination, accordez des autorisations pour la livraison des journaux d'accès au serveur à l'aide d'une liste ACL de compartiment ou d'une politique de compartiment :

   - Mettre à jour la politique de compartiment (recommandé) : pour accorder des autorisations au principal du service de journalisation, utilisez la commande `put-bucket-policy` suivante. Remplacez *`example-s3-destination-bucket-logs`* par le nom de votre compartiment de destination.

     ```
     aws s3api put-bucket-policy --bucket example-s3-destination-bucket-logs --policy
      file://policy.json
     ```

     `Policy.json` est un document JSON dans le dossier actuel qui contient la politique de compartiment suivante. Pour utiliser cette politique de compartiment, remplacez *`user input placeholders`* par vos propres informations. Dans la politique suivante, *`example-s3-destination-bucket-logs`* est le compartiment de destination où les journaux des accès au serveur seront livrés, et *`example-s3-source-bucket`* est le compartiment source. *`SOURCE-ACCOUNT-ID`* est le Compte AWS qui possède le compartiment source.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
             {
                 "Sid": "S3ServerAccessLogsPolicy",
                 "Effect": "Allow",
                 "Principal": {
                     "Service": "logging.s3.amazonaws.com"
                 },
                 "Action": [
                     "s3:PutObject"
                 ],
                 "Resource": "arn:aws:s3:::example-s3-destination-bucket-logs/*",
                 "Condition": {
                     "ArnLike": {
                         "aws:SourceArn": "arn:aws:s3:::example-s3-source-bucket"
                     },
                     "StringEquals": {
     ```

```
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Mettre à jour la liste ACL de compartiment : pour accorder des autorisations au groupe de livraison des journaux S3, utilisez la commande `put-bucket-acl` suivante. Remplacez *`example-s3-destination-bucket-logs`* par le nom de votre compartiment de destination (cible).

```
aws s3api put-bucket-acl --bucket example-s3-destination-bucket-logs  --grant-
write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp
 URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

4. Créez ensuite un fichier `logging.json` contenant votre configuration de journalisation (en vous basant sur l'un des trois exemples suivants). Après avoir créé le fichier `logging.json`, vous pouvez appliquer la configuration de journalisation à l'aide de la commande `put-bucket-logging` suivante. Remplacez *`example-s3-destination-bucket-logs`* par le nom de votre compartiment de destination (cible).

```
aws s3api put-bucket-logging --bucket example-s3-destination-bucket-logs --bucket-
logging-status file://logging.json
```

> **Note**
>
> Au lieu d'utiliser cette commande `put-bucket-logging` pour appliquer la configuration de journalisation sur chaque compartiment de destination, vous pouvez utiliser l'un des scripts bash fournis à l'étape suivante. Pour utiliser ces scripts, vous devez créer les fichiers `policy.json` et `logging.json`, comme décrit dans cette procédure.

Le fichier `logging.json` est un document JSON situé dans le dossier actuel, qui contient votre configuration de journalisation. Si un compartiment de destination utilise le paramètre

Propriétaire du compartiment appliqué pour Propriété d'objets, votre configuration de journalisation ne peut pas contenir d'octrois de destination (cibles). Pour plus d'informations, consultez Autorisations de diffusion de journaux.

Example : **`logging.json`** sans octrois de destination (cibles)

L'exemple de fichier `logging.json` suivant ne contient pas d'octrois de destination (cibles). Par conséquent, vous pouvez appliquer cette configuration à un compartiment de destination (cible) qui utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets.

```
{
    "LoggingEnabled": {
        "TargetBucket": "example-s3-destination-bucket-logs",
        "TargetPrefix": "example-s3-destination-bucket/"
     }
 }
```

Example : **`logging.json`** avec des octrois de destination (cibles)

L'exemple de fichier `logging.json` suivant contient des octrois de destination (cibles).

Si le compartiment de destination utilise le paramètre Appliqué par le propriétaire du compartiment pour Propriété d'objets, vous ne pouvez pas inclure d'octrois de destination (cibles) dans votre configuration PutBucketLogging. Pour plus d'informations, consultez Autorisations de diffusion de journaux.

```
{
    "LoggingEnabled": {
        "TargetBucket": "example-s3-destination-bucket-logs",
        "TargetPrefix": "example-s3-destination-bucket/",
        "TargetGrants": [
             {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "user@example.com"
                 },
                "Permission": "FULL_CONTROL"
             }
```

```
            ]
        }
    }
```

Example : **`logging.json`** avec le format de clé d'objet journal défini sur Heure de l'événement S3

Le fichier `logging.json` suivant remplace le format de clé d'objet journal par l'heure de l'événement S3. Pour plus d'informations sur la définition du format de clé d'objet journal, consultez the section called "Comment activer la livraison des journaux ?".

```
  {
    "LoggingEnabled": {
        "TargetBucket": "example-s3-destination-bucket-logs",
        "TargetPrefix": "example-s3-destination-bucket/",
        "TargetObjectKeyFormat": {
            "PartitionedPrefix": {
                "PartitionDateSource": "EventTime"
            }
         }
    }
}
```

5. Utilisez l'un des scripts bash suivants pour ajouter la journalisation des accès pour tous les compartiments dans votre compte. Remplacez *`example-s3-destination-bucket-logs`* par le nom de votre compartiment de destination (cible) et remplacez *`us-west-2`* par le nom de la région où se trouvent vos compartiments.

> **Note**
>
> Ce script fonctionne uniquement si tous vos compartiments se trouvent dans la même région. Si vous disposez de compartiments dans plusieurs Régions, vous devez adapter le script.

Example — Accordez l'accès avec des stratégies de compartiment et ajoutez la journalisation pour les compartiments de votre compte

```
  loggingBucket='example-s3-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" != "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json

  echo "Complete"
```

Example — Accordez l'accès avec des listes ACL de compartiment et ajoutez la journalisation pour les compartiments de votre compte

```
  loggingBucket='example-s3-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" != "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json

  echo "Complete"
```

Vérification de votre configuration des journaux d'accès au serveur

Après avoir activé la journalisation des accès au serveur, effectuez les étapes suivantes :

- Accédez au compartiment de destination et vérifiez que les fichiers journaux sont livrés. Une fois les journaux d'accès configurés, plus d'une heure peut être nécessaire pour que toutes les demandes soient correctement journalisées et livrées. Vous pouvez également vérifier automatiquement la livraison des journaux en utilisant les métriques de demande Amazon S3 et en configurant des CloudWatch alarmes Amazon pour ces métriques. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.
- Vérifiez que vous êtes en mesure d'ouvrir et de lire le contenu des fichiers journaux.

Pour obtenir des informations sur la résolution des problèmes de journalisation des accès au serveur, consultez Résolution des problèmes de journalisation des accès au serveur.

Format des journaux d'accès au serveur Amazon S3

La journalisation des accès au serveur fournit des enregistrements détaillés des demandes soumises à un compartiment Amazon S3. Vous pouvez utiliser les journaux d'accès au serveur dans les buts suivants :

- Réalisation d'audits de sécurité et d'accès
- Apprendre à connaître votre clientèle
- Comprendre votre facture Amazon S3

Cette section décrit le format et d'autres détails des fichiers journaux d'accès au serveur Amazon S3.

Les fichiers journaux d'accès au serveur consistent en une séquence d'enregistrements de journaux délimités par un retour à la ligne. Chaque enregistrement de journaux représente une demande et consiste en des champs séparés par des espaces.

Voici un exemple de journal composé de cinq enregistrements.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
```

```
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
 Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /DOC-EXAMPLE-BUCKET1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /DOC-EXAMPLE-BUCKET1?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /DOC-EXAMPLE-BUCKET1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader DOC-
EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DOC-EXAMPLE-BUCKET1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /DOC-EXAMPLE-BUCKET1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader DOC-EXAMPLE-BUCKET1.s3.us-west-1.amazonaws.com TLSV1.2
 - Yes
```

> **Note**
>
> Tous les champs peuvent être configurés sur - pour indiquer que les données étaient inconnues ou indisponibles, ou que le champ ne s'appliquait pas à cette demande.

Rubriques

- Champs d'enregistrement des journaux
- Journalisation supplémentaire les opérations de copie

- Informations personnalisées des journaux d'accès
- Remarques de programmation relatives au format étendu des journaux d'accès au serveur

Champs d'enregistrement des journaux

La liste suivante décrit les champs de l'enregistrement des journaux.

Propriétaire du compartiment

ID d'utilisateur canonique du propriétaire du compartiment source. L'ID utilisateur canonique est une autre forme de l' Compte AWS ID. Pour plus d'informations sur l'ID d'utilisateur canonique, consultez Identificateurs de Compte AWS dans Références générales AWS. Pour savoir comment trouver l'ID d'utilisateur canonique de votre compte, consultez Identification de l'ID d'utilisateur canonique de votre Compte AWS.

Exemple d'entrée

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Compartiment

Le nom du compartiment sur lequel la demande a été traitée. Si le système reçoit une demande erronée et ne peut pas déterminer le compartiment, la demande n'apparaît sur aucun journal d'accès au serveur.

Exemple d'entrée

```
DOC-EXAMPLE-BUCKET1
```

Heure

Heure à laquelle la demande a été reçue. Ces dates et heures sont exprimées en heure UTC (temps universel coordonné). Le format, en utilisant terminologie `strftime()`, est le suivant : `[%d/%b/%Y:%H:%M:%S %z]`

Exemple d'entrée

```
[06/Feb/2019:00:00:38 +0000]
```

Adresse IP distante

L'adresse IP apparente du demandeur. Les proxys et pare-feu intermédiaires doivent cacher l'adresse IP réelle de la machine à l'origine de la demande.

Exemple d'entrée

```
192.0.2.3
```

Demandeur

ID d'utilisateur canonique du demandeur, ou - pour les demandes non authentifiées. Si le demandeur était un utilisateur IAM, ce champ renvoie le nom d'utilisateur IAM du demandeur ainsi Utilisateur racine d'un compte AWS que le nom auquel appartient l'utilisateur IAM. Cet identifiant est le même que celui qui est utilisé pour contrôler l'accès.

Exemple d'entrée

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Si le demandeur utilise un rôle assumé, ce champ renvoie le rôle IAM supposé.

Exemple d'entrée

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID de demande

Chaîne de caractères générée par Amazon S3 pour identifier de façon unique chaque demande.

Exemple d'entrée

```
3E57427F33A59F07
```

Opération

L'opération listée ici est déclarée comme `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` ou `BATCH.DELETE.OBJECT`, ou `S3.action.resource_type` pour Cycle de vie et journalisation.

Exemple d'entrée

```
REST.PUT.OBJECT
```

Clé

La partie clé (nom de l'objet) de la demande.

Exemple d'entrée

```
/photos/2019/08/puppy.jpg
```

Request-URI

La partie `Request-URI` du message de requête HTTP.

Exemple d'entrée

```
"GET /DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

Statut HTTP

Code numérique du statut HTTP de la réponse.

Exemple d'entrée

```
200
```

Code d'erreur

La valeur Amazon S3 Code d'erreur ou `-` si aucune erreur ne se produit.

Exemple d'entrée

```
NoSuchBucket
```

Octets envoyés

Nombre d'octets de réponse envoyés, hors surcharge de protocole HTTP ou `-` si zéro.

Exemple d'entrée

```
2662992
```

Taille de l'objet

Taille totale de l'objet en question.

Exemple d'entrée

```
3462992
```

Durée totale

Le nombre de millisecondes (ms) pendant lesquelles la demande était en cours du point de vue du serveur. Cette valeur est mesurée entre la réception de la demande et l'envoi du dernier octet de la réponse. Les mesures effectuées depuis la perspective du client peuvent être plus longues en raison de la latence du réseau.

Exemple d'entrée

```
70
```

Délai de traitement

Le nombre de millisecondes pendant lesquelles Amazon S3 a traité la demande. Cette valeur est mesurée entre la réception du dernier octets de votre demande et l'envoi du premier octet de la réponse.

Exemple d'entrée

```
10
```

Referer

Valeur de l'en-tête du `Referer` HTTP, le cas échéant. Les agents utilisateur HTTP (par exemple, les navigateurs) définissent généralement cet en-tête comme l'URL de la page de liaison ou d'intégration lors d'une demande.

Exemple d'entrée

```
"http://www.example.com/webservices"
```

User-Agent

Valeur de l'en-tête du `User-Agent` HTTP.

Exemple d'entrée

```
"curl/7.15.1"
```

ID de version

L'ID de version dans la demande ou `-` si l'opération n'accepte pas de paramètre `versionId`.

Exemple d'entrée

```
3HL4kqtJvjVBH40Nrjfkd
```

ID de l'hôte

L'ID de la requête étendue `x-amz-id-2` ou Amazon S3.

Exemple d'entrée

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Version de signature

Version de signature, `SigV2` ou `SigV4`, qui a été utilisée pour authentifier la demande ou `-` pour les demandes non authentifiées.

Exemple d'entrée

```
SigV2
```

Suite de chiffrement

Chiffrement Secure Sockets Layer (SSL) qui a été négocié pour la requête HTTPS ou une valeur `-` pour HTTP.

Exemple d'entrée

```
ECDHE-RSA-AES128-GCM-SHA256
```

Type d'authentification

Type d'authentification de requête utilisé : `AuthHeader` pour les en-têtes d'authentification, `QueryString` pour la chaîne de requête (URL présignée), ou `-` pour les demandes non authentifiées.

Exemple d'entrée

```
AuthHeader
```

En-tête d'hôte

Point de terminaison utilisé pour vous connecter à Amazon S3.

Exemple d'entrée

```
s3.us-west-2.amazonaws.com
```

Certaines régions antérieures prennent en charge les points de terminaison hérités. Vous pouvez voir ces points de terminaison dans les journaux ou AWS CloudTrail journaux d'accès de votre serveur. Pour plus d'informations, consultez Points de terminaison hérités. Pour obtenir la liste complète des régions et points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans le Référence générale d'Amazon Web Services.

Version de TLS

Version de protocole TLS (Transport Layer Security) négociée par le client. La valeur est l'une des valeurs suivantes : `TLSv1.1`, `TLSv1.2`, `TLSv1.3` ou `-` si le protocole TLS n'a pas été utilisé.

Exemple d'entrée

```
TLSv1.2
```

ARN de point d'accès

Amazon Resource Name (ARN) du point d'accès de la demande. Si l'ARN du point d'accès est mal formé ou n'est pas utilisé, le champ contient un `-`. Pour plus d'informations sur les points

d’accès, consultez Utilisation des points d'accès. Pour plus d'informations sur les ARN, consultez Amazon Resource Name (ARN) dans le Guide de référence générale AWS .

Exemple d'entrée

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Une chaîne qui indique si la requête a nécessité une liste de contrôle d'accès (ACL) pour l'autorisation. Si la requête a nécessité un ACL pour l'autorisation, la chaîne est `Yes`. Si aucune ACL n'est requise, la chaîne est `-`. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL). Pour plus d'informations sur l'utilisation du champ `aclRequired` pour désactiver les ACL, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Exemple d'entrée

```
Yes
```

Journalisation supplémentaire les opérations de copie

Une copie implique une demande `GET` et une demande `PUT`. C'est pourquoi, nous consignons deux enregistrements lors d'une opération de copie. La section précédente décrit les champs liés à la partie `PUT` de l'opération. La liste suivante décrit les champs dans l'enregistrement qui ont trait à la partie `GET` de l'opération de copie.

Propriétaire du compartiment

ID d'utilisateur canonique du compartiment qui stocke l'objet à copier. L'ID utilisateur canonique est une autre forme de l' Compte AWS ID. Pour plus d'informations sur l'ID d'utilisateur canonique, consultez Identificateurs de Compte AWS dans Références générales AWS. Pour savoir comment trouver l'ID d'utilisateur canonique de votre compte, consultez Identification de l'ID d'utilisateur canonique de votre Compte AWS.

Exemple d'entrée

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Compartiment

Nom du compartiment qui stocke l'objet à copier.

Exemple d'entrée

```
DOC-EXAMPLE-BUCKET1
```

Heure

Heure à laquelle la demande a été reçue. Ces dates et heures sont exprimées en heure UTC (temps universel coordonné). Le format, en utilisant terminologie `strftime()`, est le suivant : `[%d/%B/%Y:%H:%M:%S %z]`

Exemple d'entrée

```
[06/Feb/2019:00:00:38 +0000]
```

Adresse IP distante

L'adresse IP apparente du demandeur. Les proxys et pare-feu intermédiaires doivent cacher l'adresse IP réelle de la machine à l'origine de la demande.

Exemple d'entrée

```
192.0.2.3
```

Demandeur

ID d'utilisateur canonique du demandeur, ou - pour les demandes non authentifiées. Si le demandeur était un utilisateur IAM, ce champ renverra le nom d'utilisateur IAM du demandeur ainsi Utilisateur racine d'un compte AWS que le nom auquel appartient l'utilisateur IAM. Cet identifiant est le même que celui qui est utilisé pour contrôler l'accès.

Exemple d'entrée

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Si le demandeur utilise un rôle assumé, ce champ renvoie le rôle IAM supposé.

Exemple d'entrée

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

ID de demande

Chaîne de caractères générée par Amazon S3 pour identifier de façon unique chaque demande.

Exemple d'entrée

```
3E57427F33A59F07
```

Opération

L'opération listée ici est déclarée comme `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` ou `BATCH.DELETE.OBJECT`.

Exemple d'entrée

```
REST.COPY.OBJECT_GET
```

Clé

La clé (nom de l'objet) de l'objet à copier, ou `-` si l'opération n'accepte pas de paramètre clé.

Exemple d'entrée

```
/photos/2019/08/puppy.jpg
```

Request-URI

La partie `Request-URI` du message de requête HTTP.

Exemple d'entrée

```
"GET /DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-foo=bar"
```

Statut HTTP

Code numérique du statut HTTP de la partie `GET` de l'opération de copie.

Exemple d'entrée

```
200
```

Code d'erreur

La valeur Amazon S3 Code d'erreur de la partie GET de l'opération de copie ou - si aucune erreur ne se produit.

Exemple d'entrée

```
NoSuchBucket
```

Octets envoyés

Nombre d'octets de réponse envoyés, hors surcharge de protocole HTTP ou - si zéro.

Exemple d'entrée

```
2662992
```

Taille de l'objet

Taille totale de l'objet en question.

Exemple d'entrée

```
3462992
```

Durée totale

Le nombre de millisecondes (ms) pendant lesquelles la demande était en cours du point de vue du serveur. Cette valeur est mesurée entre la réception de la demande et l'envoi du dernier octet de la réponse. Les mesures effectuées depuis la perspective du client peuvent être plus longues en raison de la latence du réseau.

Exemple d'entrée

```
70
```

Délai de traitement

Le nombre de millisecondes pendant lesquelles Amazon S3 a traité la demande. Cette valeur est mesurée entre la réception du dernier octets de votre demande et l'envoi du premier octet de la réponse.

Exemple d'entrée

```
10
```

Referer

Valeur de l'en-tête du `Referer` HTTP, le cas échéant. Les agents utilisateur HTTP (par exemple, les navigateurs) définissent généralement cet en-tête comme l'URL de la page de liaison ou d'intégration lors d'une demande.

Exemple d'entrée

```
"http://www.example.com/webservices"
```

User-Agent

Valeur de l'en-tête du `User-Agent` HTTP.

Exemple d'entrée

```
"curl/7.15.1"
```

ID de version

ID de version de l'objet à copier ou `-` si l'en-tête `x-amz-copy-source` n'a pas spécifié de paramètre `versionId` dans la source de copie.

Exemple d'entrée

```
3HL4kqtJvjVBH40Nrjfkd
```

ID de l'hôte

L'ID de la requête étendue `x-amz-id-2` ou Amazon S3.

Exemple d'entrée

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Version de signature

Version de signature, `SigV2` ou `SigV4`, qui a été utilisée pour authentifier la requête ou `-` pour les demandes non authentifiées.

Exemple d'entrée

```
SigV4
```

Suite de chiffrement

Chiffrement Secure Sockets Layer (SSL) qui a été négocié pour la requête HTTPS, ou `-` pour HTTP.

Exemple d'entrée

```
ECDHE-RSA-AES128-GCM-SHA256
```

Type d'authentification

Type d'authentification de requête utilisé : `AuthHeader` pour les en-têtes d'authentification, `QueryString` pour les chaînes de requête (URL présignée), ou `-` pour les demandes non authentifiées.

Exemple d'entrée

```
AuthHeader
```

En-tête d'hôte

Point de terminaison utilisé pour vous connecter à Amazon S3.

Exemple d'entrée

```
s3.us-west-2.amazonaws.com
```

Certaines régions antérieures prennent en charge les points de terminaison hérités. Vous pouvez voir ces points de terminaison dans les journaux ou AWS CloudTrail journaux d'accès de votre serveur. Pour plus d'informations, consultez Points de terminaison hérités. Pour obtenir la liste

complète des régions et points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans le Référence générale d'Amazon Web Services.

Version de TLS

Version de protocole TLS (Transport Layer Security) négociée par le client. La valeur est l'une des valeurs suivantes : `TLSv1.1`, `TLSv1.2`, `TLSv1.3` ou `-` si le protocole TLS n'a pas été utilisé.

Exemple d'entrée

```
TLSv1.2
```

ARN de point d'accès

Amazon Resource Name (ARN) du point d'accès de la demande. Si l'ARN du point d'accès est mal formé ou n'est pas utilisé, le champ contient un `-`. Pour plus d'informations sur les points d'accès, consultez Utilisation des points d'accès. Pour plus d'informations sur les ARN, consultez Amazon Resource Name (ARN) dans le Guide de référence générale AWS .

Exemple d'entrée

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Une chaîne qui indique si la requête a nécessité une liste de contrôle d'accès (ACL) pour l'autorisation. Si la requête a nécessité un ACL pour l'autorisation, la chaîne est `Yes`. Si aucune ACL n'est requise, la chaîne est `-`. Pour en savoir plus sur les listes ACL, consultez Présentation de la liste de contrôle d'accès (ACL). Pour plus d'informations sur l'utilisation du champ `aclRequired` pour désactiver les ACL, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Exemple d'entrée

```
Yes
```

Informations personnalisées des journaux d'accès

Vous pouvez inclure des informations personnalisées à stocker dans le journal d'accès pour une demande. Pour ce faire, ajoutez un paramètre de chaîne de requête personnalisé à l'URL utilisée.

Amazon S3 ignore les paramètres de la chaîne de requête qui commencent par `x-`, mais inclut ces derniers dans les enregistrements des journaux d'accès pour la demande dans le champ `Request-URI`.

Par exemple, une demande `GET` pour `"s3.amazonaws.com/DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg?x-user=johndoe"` fonctionne de la même manière que la demande pour `"s3.amazonaws.com/DOC-EXAMPLE-BUCKET1/photos/2019/08/puppy.jpg"`, sauf que la chaîne `"x-user=johndoe"` est incluse dans le champ `Request-URI` de l'enregistrement de journal associé. Cette fonctionnalité est uniquement disponible dans l'interface REST.

Remarques de programmation relatives au format étendu des journaux d'accès au serveur

Parfois, nous pouvons étendre le format d'enregistrement du journal d'accès en ajoutant de nouveaux champs à la fin de chaque ligne. Par conséquent, assurez-vous que tout votre code qui analyse les journaux d'accès au serveur peut gérer les champs de suivi qu'il pourrait ne pas comprendre.

Suppression des fichiers journaux Amazon S3

Un compartiment Amazon S3 dont la journalisation des accès au serveur est activée peut accumuler un grand nombre de journaux au fil du temps. L'application peut avoir besoin de ces journaux d'accès pendant une période donnée après leur création, après quoi vous pourrez les supprimer. Vous pouvez utiliser la configuration de cycle de vie Amazon S3 pour définir des règles afin qu'Amazon S3 mette automatiquement ces objets en file d'attente de suppression à la fin de leur vie.

Vous pouvez définir une configuration de cycle de vie pour un sous-ensemble d'objets dans votre compartiment S3 à l'aide d'un préfixe partagé. Si vous spécifiez un préfixe dans la configuration de la journalisation des accès au serveur, vous pouvez configurer une règle de configuration du cycle de vie pour supprimer les objets du journal qui ont ce préfixe.

Par exemple, supposons que vos objets journaux aient le préfixe `logs/`. Vous pouvez définir une règle de configuration de cycle de vie pour supprimer tous les objets du compartiment dotés du préfixe `logs/` après une période de temps spécifiée.

Pour en savoir plus sur la configuration du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

Pour plus d'informations sur la journalisation des accès au serveur, consultez Enregistrement de demandes avec journalisation des accès au serveur.

Utilisation des journaux d’accès au serveur Amazon S3 pour identifier des demandes

Vous pouvez identifier des demandes Amazon S3 à l’aide des journaux d’accès au serveur Amazon S3.

> **Note**
>
> - Pour identifier les demandes Amazon S3, nous vous recommandons d'utiliser AWS CloudTrail des événements de données plutôt que des journaux d'accès au serveur Amazon S3. CloudTrail les événements de données sont plus faciles à configurer et contiennent davantage d'informations. Pour plus d’informations, consultez Identification des demandes Amazon S3 à l'aide CloudTrail.
> - Selon le nombre de demandes d'accès que vous recevez, l'analyse de vos journaux peut nécessiter plus de ressources ou de temps que l'utilisation d'événements de CloudTrail données.

Rubriques

Interrogation des journaux d’accès pour les demandes à l’aide d’Amazon Athena

Vous pouvez identifier les demandes Amazon S3 à l’aide des journaux d’accès Amazon S3 en utilisant Amazon Athena.

Amazon S3 stocke les journaux d'accès au serveur en tant qu'objets dans un compartiment S3. Il est souvent plus facile d'utiliser un outil capable d'analyser les journaux dans Amazon S3. Athena prend en charge l'analyse des objets S3 et ne peut pas être utilisé pour interroger les journaux d'accès Amazon S3.

Example

L'exemple suivant montre comment vous pouvez interroger les journaux d'accès au serveur Amazon S3 dans Amazon Athena. Remplacez les *user input placeholders* utilisés dans les exemples suivants par vos propres informations.

> **Note**
>
> Pour spécifier un emplacement Amazon S3 dans une requête Athena, vous devez fournir un URI S3 pour le compartiment où vos journaux sont livrés. Cet URI doit inclure le nom et le préfixe du compartiment au format suivant : `s3://`*example-s3-bucket1*`-logs/`*prefix*`/`

1. Ouvrez la console Athena à l'adresse https://console.aws.amazon.com/athena/.
2. Dans l'Éditeur de requête, exécutez une commande similaire à ce qui suit. Remplacez *s3_access_logs_db* par le nom que vous souhaitez donner à votre base de données.

```
CREATE DATABASE s3_access_logs_db
```

> **Note**
>
> Il est recommandé de créer la base de données au même endroit Région AWS que votre compartiment S3.

3. Dans l'Éditeur de requête, exécutez une commande similaire à ce qui suit pour créer un schéma de table dans la base de données que vous avez créée à l'étape 2. Remplacez *s3_access_logs_db.mybucket_logs* par le nom que vous souhaitez donner à votre table. Les valeurs de type de données `STRING` et `BIGINT` sont les propriétés des journaux d'accès. Vous pouvez interroger ces propriétés dans Athena. Pour `LOCATION`, saisissez le compartiment S3 et le préfixe du chemin comme notés précédemment.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
```

```
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING,
  `accesspointarn` STRING,
  `aclrequired` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://DOC-EXAMPLE-BUCKET1-logs/prefix/'
```

4. Dans le volet de navigation, sous Database (Base de données), choisissez votre base de données.
5. Sous Tables, choisissez Aperçu de la table en regard du nom de votre table.

   Dans le volet Results (Résultats), vous devriez voir apparaître les données des journaux d'accès au serveur, par exemple `bucketowner`, `bucket`, `requestdatetime`, etc. Ceci signifie que vous avez correctement créé la table Athena. Vous pouvez désormais interroger les journaux d'accès au serveur Amazon S3.

Example – Afficher qui a supprimé un objet et quand (horodatage, adresse IP et utilisateur IAM)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example – Afficher toutes les opérations effectuées par un utilisateur IAM

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example – Afficher toutes les opérations effectuées sur un objet au cours d'une période spécifique

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
    BETWEEN parse_datetime('2017-02-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
    AND parse_datetime('2017-02-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

Example — afficher la quantité de données transférées vers une adresse IP donnée au cours d'une période

```
SELECT coalesce(SUM(bytessent), 0) AS bytessenttotal
FROM s3_access_logs_db.mybucket_logs
WHERE remoteip='192.0.2.1'
AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-06-01','yyyy-MM-dd')
AND parse_datetime('2022-07-01','yyyy-MM-dd');
```

> **Note**
>
> Pour réduire le temps de conservation de vos journaux, vous pouvez créer une configuration de cycle de vie S3 pour votre compartiment de journaux d'accès au serveur. Créez des règles de configuration de cycle de vie pour supprimer régulièrement les fichiers journaux. Cela permet de réduire la quantité de données analysées par Athena pour chaque requête. Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket.

Identification des demandes Signature Version 2 à l'aide des journaux d'accès Amazon S3

La prise en charge d'Amazon S3 pour Signature Version 2 sera désactivée (obsolète). Après cela, Amazon S3 n'acceptera plus les demandes utilisant Signature Version 2, et toutes les demandes devront utiliser le processus de signature Signature Version 4. Vous pouvez identifier les demandes d'accès avec Signature Version 2 en utilisant les journaux d'accès Amazon S3.

> **Note**
>
> Pour identifier les demandes Signature version 2, nous vous recommandons d'utiliser AWS CloudTrail des événements de données plutôt que des journaux d'accès au serveur Amazon S3. CloudTrail les événements de données sont plus faciles à configurer et contiennent plus d'informations que les journaux d'accès au serveur. Pour plus d'informations, consultez Identification des demandes Amazon S3 Signature version 2 à l'aide de CloudTrail.

Example – Afficher tous les demandeurs qui envoient du trafic Signature Version 2

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Identification des demandes d’accès aux objets à l’aide des journaux d’accès Amazon S3

Vous pouvez utiliser des requêtes sur les journaux d’accès au serveur Amazon S3 pour identifier les demandes d’accès aux objets Amazon S3, pour des opérations telles que `GET`, `PUT` et `DELETE`, et découvrir plus d’informations sur ces demandes.

L’exemple de requête Amazon Athena suivant montre comment obtenir toutes les demandes d’objet `PUT` pour Amazon S3 à partir d’un journal d’accès au serveur.

Example — afficher tous les demandeurs qui envoient des demandes d'objets **PUT** au cours d'une période donnée

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

L'exemple de requête Amazon Athena suivant montre comment obtenir toutes les demandes d'objet `GET` pour Amazon S3 à partir du journal d'accès au serveur.

Example — afficher tous les demandeurs qui envoient des demandes d'objets **GET** au cours d'une période donnée

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

L'exemple de requête Amazon Athena suivant montre comment obtenir toutes les demandes anonymes vers vos compartiments S3 à partir du journal d'accès du serveur.

Example — afficher tous les demandeurs anonymes qui adressent des demandes à un compartiment au cours d'une période donnée

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db.mybucket_logs
WHERE requester IS NULL AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

La requête Amazon Athena suivante montre comment identifier toutes les requêtes adressées à vos compartiments S3 qui ont nécessité une liste de contrôle d'accès (ACL) pour l'autorisation. Vous pouvez utiliser ces informations pour migrer ces autorisations ACL vers les politiques de compartiment appropriées et désactiver les ACL. Après avoir créé ces politiques de compartiment, vous pouvez désactiver les ACL pour ces compartiments. Pour plus d'informations sur la désactivation des ACL, consultez Conditions préalables à la désactivation des listes ACL.

Example — identifiez toutes les demandes qui ont nécessité un ACL pour l'autorisation.

```
SELECT bucket_name, requester, key, operation, aclrequired, requestdatetime
FROM s3_access_logs_db
WHERE aclrequired = 'Yes' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-05-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2022-08-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
```

Note

- Vous pouvez modifier la plage de dates en fonction de vos besoins.
- Ces exemples de requêtes peuvent aussi s'avérer utiles pour surveiller la sécurité. Vous pouvez vérifier les résultats pour les appels `PutObject` ou `GetObject` depuis des adresses IP ou des demandeurs inattendus ou non autorisés et pour identifier les demandes anonymes adressées à vos compartiments.
- Cette requête ne récupère d'informations qu'à partir du moment où l'enregistrement a été activé.

- Si vous utilisez AWS CloudTrail des journaux, consultezIdentification de l'accès aux objets S3 en utilisant CloudTrail.

Surveillance des métriques avec Amazon CloudWatch

CloudWatch Les métriques Amazon pour Amazon S3 peuvent vous aider à comprendre et à améliorer les performances des applications qui utilisent Amazon S3. Il existe plusieurs méthodes que vous pouvez utiliser CloudWatch avec Amazon S3.

Métriques du stockage quotidien pour les compartiments

Surveillez le stockage par compartiments à l'aide CloudWatch de ce qui collecte et traite les données de stockage d'Amazon S3 en indicateurs quotidiens lisibles. Ces métriques de stockage pour Amazon S3 sont fournis une fois par jour à tous les clients sans coût supplémentaire.

Métriques des demandes

Surveillez les demandes Amazon S3 pour identifier rapidement les problèmes opérationnels et agir en conséquences. Les métriques sont disponibles à des intervalles d'une minute après une latence de traitement. Ces CloudWatch statistiques sont facturées au même tarif que les statistiques CloudWatch personnalisées d'Amazon. Pour plus d'informations sur CloudWatch les tarifs, consultez CloudWatch les tarifs Amazon. Pour en savoir plus sur la façon d'obtenir ces métriques, veuillez consulter CloudWatch configurations de métriques.

Lorsqu'elles sont activées, les métriques de demandes sont indiquées pour toutes les opérations d'objets. Par défaut, ces métriques d'une minute sont disponibles au niveau du compartiment Amazon S3. Vous pouvez également définir un filtre pour les métriques à l'aide d'un préfixe partagé, d'une balise d'objet ou d'un point d'accès.

- Point d'accès – Les points d'accès sont nommés points de terminaison réseau qui sont associés à des compartiments et simplifient la gestion de l'accès aux données à grande échelle pour les jeux de données partagés dans S3. Avec le filtre de point d'accès, vous pouvez obtenir des informations sur l'utilisation de votre point d'accès. Pour plus d'informations sur les points d'accès, consultez Surveillance et journalisation des points d'accès.
- Préfixe – Même si le modèle de données Amazon S3 est une structure plane, vous pouvez induire une hiérarchie à l'aide d'un préfixe. Un préfixe est similaire à un nom de répertoire qui vous permet de regrouper des objets similaires dans un compartiment. La console S3 prend en charge ces préfixes avec le concept de dossiers. Si vous filtrez par préfixe, les objets ayant

le même préfixe seront inclus dans la configuration des métriques. Pour en savoir plus sur les préfixes, consultez Organisation des objets à l'aide de préfixes.

- Balises – Vous pouvez ajouter des balises, à savoir des paires de noms clés-valeurs, aux objets. Les balises vous permettent de trouver et de classer facilement des objets. Vous pouvez également utiliser des balises comme filtre pour les configurations de métriques de sorte que seuls les objets avec ces balises soient inclus dans la configuration des métriques. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.

Pour faire correspondre ces métriques à des applications commerciales spécifiques, à des flux de travail ou à des organisation internes spécifiques, vous pouvez filtrer sur un préfixe partagé, une balise

Métriques de réplication

Métriques de réplication : surveillez le nombre total d'opérations d'API S3 en attente de réplication, la taille totale des objets en attente de réplication et la durée maximale de réplication vers la Région AWS de destination, et le nombre total d'opérations pour lesquelles la réplication a échoué. Les règles de réplication sur lesquelles des métriques de réplication S3 ou Contrôle du temps de réplication S3 (S3 RTC) sont activées publient des métriques de réplication.

Pour plus d'informations, consultez Surveillance de la progression avec des métriques de réplication et des notifications d'événements S3 ou Satisfaire aux exigences de conformité à l'aide du contrôle du délai de réplication S3 (S3 RTC).

Métriques Amazon S3 Storage Lens

Vous pouvez publier les statistiques d'utilisation et d'activité de S3 Storage Lens sur Amazon CloudWatch afin de créer une vue unifiée de votre santé opérationnelle dans des CloudWatch tableaux de bord. Les métriques S3 Storage Lens sont disponibles dans l'espace de noms `AWS/S3/Storage-Lens`. L'option de CloudWatch publication est disponible pour les tableaux de bord S3 Storage Lens mis à niveau vers des métriques et des recommandations avancées. Vous pouvez activer l'option de CloudWatch publication pour une configuration de tableau de bord nouvelle ou existante dans S3 Storage Lens.

Pour plus d'informations, consultez Surveillance des métriques S3 Storage Lens dans CloudWatch.

Toutes les CloudWatch statistiques sont conservées pendant une période de 15 mois afin que vous puissiez accéder aux informations historiques et avoir une meilleure idée des performances de votre

application ou service Web. Pour plus d'informations CloudWatch, consultez Qu'est-ce qu'Amazon CloudWatch ? dans le guide de CloudWatch l'utilisateur Amazon. Vous aurez peut-être besoin de configurations supplémentaires pour vos CloudWatch alarmes, en fonction de vos cas d'utilisation. Par exemple, vous pouvez utiliser une expression mathématique métrique pour créer une alarme. Pour plus d'informations, consultez CloudWatch les sections Utiliser des métriques, Utiliser des métriques mathématiques, Utiliser des CloudWatch alarmes Amazon et Créer une CloudWatch alarme basée sur une expression mathématique métrique dans le Guide de CloudWatch l'utilisateur Amazon.

Livraison des CloudWatch indicateurs les plus efficaces

CloudWatch les indicateurs sont fournis dans la mesure du possible. La plupart des demandes relatives à un objet Amazon S3 comportant des métriques de demande aboutissent à l'envoi d'un point de données à CloudWatch.

L'exhaustivité et la ponctualité des métriques ne sont pas garanties. Le point de données d'une demande particulière doit être retourné avec un horodatage ultérieur au moment du traitement de la demande. Le point de données peut être retardé d'une minute avant d'être disponible CloudWatch, ou il se peut qu'il ne soit pas livré du tout. CloudWatch les métriques de demande vous donnent une idée de la nature du trafic par rapport à votre compartiment en temps quasi réel. L'objectif n'est pas de comptabiliser toutes les demandes.

En conséquence de la nature du « meilleur effort » de cette fonction, les rapports disponibles sur le Tableau de bord Gestion de la facturation et des coûts peuvent inclure une ou plusieurs demandes d'accès qui ne figurent pas dans les métriques du compartiment.

Pour plus d’informations, consultez les rubriques suivantes.

Rubriques

- Métriques et dimensions
- Accès aux CloudWatch métriques
- CloudWatch configurations de métriques

Métriques et dimensions

Les métriques et dimensions de stockage qu'Amazon S3 envoie à Amazon CloudWatch sont répertoriées dans les tableaux suivants.

Livraison des CloudWatch indicateurs les plus efficaces

CloudWatch les indicateurs sont fournis dans la mesure du possible. La plupart des demandes relatives à un objet Amazon S3 comportant des métriques de demande aboutissent à l'envoi d'un point de données à CloudWatch.

L'exhaustivité et la ponctualité des métriques ne sont pas garanties. Le point de données d'une demande particulière doit être retourné avec un horodatage ultérieur au moment du traitement de la demande. Le point de données peut être retardé d'une minute avant d'être disponible CloudWatch, ou il se peut qu'il ne soit pas livré du tout. CloudWatch les métriques de demande vous donnent une idée de la nature du trafic par rapport à votre compartiment en temps quasi réel. L'objectif n'est pas de comptabiliser toutes les demandes.

En conséquence de la nature du « meilleur effort » de cette fonction, les rapports disponibles sur le Tableau de bord Gestion de la facturation et des coûts peuvent inclure une ou plusieurs demandes d'accès qui ne figurent pas dans les métriques du compartiment.

Rubriques

Mesures de stockage quotidiennes d'Amazon S3 pour les compartiments CloudWatch

L'espace de nom `AWS/S3` inclut les métriques quotidiennes de stockage suivantes pour les compartiments.

Métrique	Description
`BucketSizeBytes`	Quantité de données en octets stockée dans un compartiment dans les classes de stockage suivantes :

Métrique	Description
	• S3 Standard (STANDARD) • S3 – Hiérarchisation intelligente (INTELLIGENT_TIERING) • S3 Standard – Accès peu fréquent (STANDARD_IA) • Accès peu fréquent à S3 One Zone () ONEZONE_IA • Stockage à redondance réduite (RRS) (REDUCED_REDUNDANCY) • S3 Glacier Instant Retrieval (GLACIER_IR) • S3 Glacier Deep Archive (DEEP_ARCHIVE) • S3 Glacier Flexible Retrieval (GLACIER) • S3 Express One Zone (EXPRESS_ONEZONE) Cette valeur est calculée en additionnant la taille de tous les objets et métadonnées (tels que les noms de bucket) du bucket (objets actuels et non courants), y compris la taille de toutes les parties pour tous les téléchargements partitionnés incomplets vers le bucket. Filtres de type de stockage valides : StandardStorage , IntelligentTieringFAStorage , IntelligentTieringIAStorage , IntelligentTieringAAStorage , IntelligentTieringAIAStorage , IntelligentTieringDAAStorage , StandardIAStorage , StandardIASizeOverhead , StandardIAObjectOverhead , OneZoneIAStorage , OneZoneIASizeOverhead , ReducedRedundancyStorage , GlacierInstantRetrievalSizeOverhead GlacierInstantRetrievalStorage , GlacierStorage , GlacierStagingStorage , GlacierObjectOverhead , GlacierS3ObjectOverhead , DeepArchiveStorage , DeepArchiveObjectOverhead , DeepArchiveS3ObjectOverhead , DeepArchiveStagingStorage et ExpressOneZone (voir la dimension StorageType) Unités : octets Statistiques valides : Moyenne

Métrique	Description
`NumberOfObjects`	Nombre total d'objets stockés dans un compartiment à usage général pour toutes les classes de stockage. Cette valeur est calculée en comptant tous les objets au sein du compartiment, ce qui inclut les versions actuelles et anciennes des objets, les marqueurs de suppressi on, ainsi que le nombre total de parties pour tous les chargements partitionnés incomplets vers le compartiment. Pour les compartiments de répertoire contenant des objets de la classe de stockage S3 Express One Zone, cette valeur est calculée en comptant tous les objets du compartiment, mais elle n'inclut pas les téléchargements multiples incomplets vers le compartiment. Filtres de type de stockage valides : `AllStorageTypes` (voir la dimension `StorageType`) Unités : nombre Statistiques valides : Moyenne

Mesures de demande Amazon S3 dans CloudWatch

L'espace de nom `AWS/S3` inclut les métriques de demandes suivantes. Ces statistiques incluent les demandes non facturables (dans le cas des `GET` demandes provenant de `CopyObject` et de réplication).

> **Note**
>
> Les métriques de requêtes Amazon S3 entrées CloudWatch ne sont pas prises en charge pour les compartiments d'annuaire.

Métrique	Description
`AllRequests`	Nombre total de demandes HTTP à destination d'un compartiment Amazon S3, quel que soit leur type. Si vous utilisez une configuration de

Métrique	Description
	métriques avec un filtre, cette métrique retourne uniquement les requêtes HTTP qui correspondent aux exigences du filtre. Unités : nombre Statistiques valides : somme
`GetRequests`	Nombre de requêtes HTTP `GET` exécutées pour des objets dans un compartiment Amazon S3. Cela n'inclut pas les opérations de liste. Cette métrique est incrémentée pour la source de chaque `CopyObject` demande. Unités : nombre Statistiques valides : somme **Note** Les demandes orientées vers des listes paginées, telles que ListMultipartUploads, ListParts, et autres ListObjectVersions, ne sont pas incluses dans cette métrique.
`PutRequests`	Nombre de requêtes HTTP `PUT` exécutées pour des objets dans un compartiment Amazon S3. Cette métrique est incrémentée pour la destination de chaque `CopyObject` demande. Unités : nombre Statistiques valides : somme
`DeleteRequests`	Nombre de requêtes HTTP `DELETE` exécutées pour des objets dans un compartiment Amazon S3. Cette métrique inclut également les DeleteObjectsdemandes. Cette métrique indique le nombre de demandes effectuées, et non pas le nombre d'objets supprimés. Unités : nombre Statistiques valides : somme

Métrique	Description
`HeadRequests`	Nombre de requêtes HTTP `HEAD` effectuées pour un compartiment Amazon S3. Unités : nombre Statistiques valides : somme
`PostRequests`	Nombre de requêtes HTTP `POST` effectuées pour un compartiment Amazon S3. Unités : nombre Statistiques valides : somme **Note** DeleteObjectset les SelectObjectContentdemandes ne sont pas incluses dans cette métrique.
`SelectRequests`	Le nombre de SelectObjectContentdemandes Amazon S3 effectuées pour des objets d'un compartiment Amazon S3. Unités : nombre Statistiques valides : somme
`SelectBytesScanned`	Le nombre d'octets de données analysés avec des SelectObjectContentrequêtes Amazon S3 dans un compartiment Amazon S3. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9

Métrique	Description
`SelectBytesReturned`	Le nombre d'octets de données renvoyés avec les SelectObjectContent requêtes Amazon S3 dans un compartiment Amazon S3. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9
`ListRequests`	Le nombre de requêtes HTTP qui répertorient le contenu d'un compartiment. Unités : nombre Statistiques valides : somme
`BytesDownloaded`	Nombre d'octets téléchargés pour les demandes adressées à un compartiment Amazon S3, pour lesquelles la réponse inclut un corps. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9
`BytesUploaded`	Nombre d'octets téléchargés pour les requêtes adressées à un compartiment Amazon S3, pour lesquelles la réponse inclut un corps. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9

Métrique	Description
`4xxErrors`	Le nombre de demandes de code d'état d'erreur du client HTTP 4 xx adressées à un compartiment Amazon S3 avec une valeur de 0 ou 1. La statistique Average (Moyenne) indique le taux d'erreur et la statistiq ue Sum (Somme) indique le nombre de ce type d'erreur pendant chaque période. Unités : nombre Statistiques valides : Moyenne (rapports par demande), Somme (rapports par période), Min, Max, Exemple de comptage
`5xxErrors`	Le nombre de demandes de code d'état d'erreur du serveur HTTP 5 xx adressées à un compartiment Amazon S3 avec une valeur de 0 ou 1. La statistique Average (Moyenne) indique le taux d'erreur et la statistiq ue Sum (Somme) indique le nombre de ce type d'erreur pendant chaque période. Unités : nombre Statistiques valides : Moyenne (rapports par demande), Somme (rapports par période), Min, Max, Exemple de comptage
`FirstByte Latency`	Temps par demande entre la réception de la demande complète par un compartiment Amazon S3 et le début du retour de la réponse. Unités : millisecondes Statistiques valides : Average (Moyenne), Sum (Somme), Min, Max (identique à p100), Sample Count (Nombre d'exemples), tout centile entre p0.0 et p100

Métrique	Description
`TotalRequestLatency`	Temps par demande écoulé entre la réception du premier octet et l'envoi du dernier octet à un compartiment Amazon S3. Cette métrique inclut le temps mis pour recevoir le corps de la requête et envoyer le corps de la réponse, qui n'est pas inclus dans `FirstByteLatency` . Unités : millisecondes Statistiques valides : Average (Moyenne), Sum (Somme), Min, Max (identique à p100), Sample Count (Nombre d'exemples), tout centile entre p0.0 et p100

Métriques de réplication S3 dans CloudWatch

Vous pouvez surveiller l'avancement de la réplication avec les métriques de réplication S3 en suivant les octets en attente, les opérations en attente et la latence de réplication. Pour plus d'informations, consultez Surveillance de l'avancement des métriques de réplication.

> **Note**
>
> Vous pouvez activer les alarmes pour vos métriques de réplication dans Amazon CloudWatch. Lorsque vous configurez des alarmes pour vos métriques de réplication, définissez le champ Missing data treatment (Traitement des données manquantes) sur Treat missing data as ignore (maintain the alarm state) (Ignorer les données manquantes [conserver l'état d'alarme]).

Métrique	Description
`ReplicationLatency`	Nombre maximal de secondes pendant lequel la destination de réplication se Région AWS trouve derrière la source Région AWS pour une règle de réplication donnée. Unités : secondes Statistiques valides : Max

Métrique	Description
`BytesPendingReplication`	Nombre total d'octets d'objets en attente de réplication pour une règle de réplication donnée. Unités : octets Statistiques valides : Max
`OperationsPendingReplication`	Nombre d'opérations en attente de réplication pour une règle de réplication donnée. Unités : nombre Statistiques valides : Max
`OperationsFailedReplication`	Nombre d'opérations pour lesquelles la réplication a échoué pour une règle de réplication donnée. Unités : nombre Statistiques valides : somme (nombre total d'opérations ayant échoué), moyenne (taux d'échec), nombre d'échantillons (nombre total d'opérations de réplication)

Métriques de S3 Storage Lens dans CloudWatch

Vous pouvez publier les indicateurs d'utilisation et d'activité de S3 Storage Lens sur Amazon CloudWatch afin de créer une vue unifiée de votre santé opérationnelle dans des CloudWatchtableaux de bord. Les métriques S3 Storage Lens sont publiées dans l'espace de `AWS/S3/Storage-Lens` noms dans CloudWatch. L'option de CloudWatch publication est disponible pour les tableaux de bord S3 Storage Lens qui ont été mis à niveau vers des métriques et des recommandations avancées.

Pour obtenir la liste des métriques de S3 Storage Lens publiées sur CloudWatch, consultezGlossaire des métriques Amazon S3 Storage Lens. Pour obtenir la liste complète des dimensions, veuillez consulter Dimensions.

Métriques de demande Lambda d'un objet S3 dans CloudWatch

S3 Object Lambda inclut les métriques de demande suivantes.

Métrique	Description
`AllRequests`	Nombre total de demandes HTTP exécutées sur un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda. Unités : nombre Statistiques valides : somme
`GetRequests`	Nombre de demandes HTTP `GET` exécutées pour des objets à l'aide d'un point d'accès Object Lambda. Cette métrique n'inclut pas les opérations de liste. Unités : nombre Statistiques valides : somme
`BytesUploaded`	Nombre d'octets chargés dans un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda, où la demande inclut un corps. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9
`PostRequests`	Nombre de demandes HTTP `POST` exécutées sur un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda. Unités : nombre Statistiques valides : somme
`PutRequests`	Nombre de demandes HTTP `PUT` exécutées pour des objets dans un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda. Unités : nombre

Métrique	Description
	Statistiques valides : somme
`DeleteRequests`	Nombre de demandes HTTP `DELETE` exécutées pour des objets dans un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda. Cette métrique inclut les DeleteObjectsdemandes. Cette métrique indique le nombre de demandes effectuées, et non pas le nombre d'objets supprimés. Unités : nombre Statistiques valides : somme
`BytesDownloaded`	Nombre d'octets téléchargés pour les demandes adressées à un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda, pour lesquelles la réponse inclut un corps. Unités : octets Statistiques valides : Moyenne (octets par demande), Somme (octets par période), Exemple de comptage, Min, Max (identique à p100), tout percentile entre p0.0 et p99.9
`FirstByte Latency`	Temps par demande entre la réception de la demande complète par un compartiment Amazon S3 via un point d'accès Object Lambda et le début du renvoi de la réponse. Cette métrique dépend du temps d'exécution de la fonction AWS Lambda pour transformer l'objet avant que la fonction ne renvoie les octets au point d'accès Object Lambda. Unités : millisecondes Statistiques valides : Average (Moyenne), Sum (Somme), Min, Max (identique à p100), Sample Count (Nombre d'exemples), tout centile entre p0.0 et p100

Métrique	Description
`TotalRequestLatency`	Temps par demande écoulé entre la réception du premier octet et l'envoi du dernier octet à un point d'accès Object Lambda. Cette métrique inclut le temps mis pour recevoir le corps de la demande et envoyer le corps de la réponse, qui n'est pas inclus dans `FirstByteLatency` . Unités : millisecondes Statistiques valides : Average (Moyenne), Sum (Somme), Min, Max (identique à p100), Sample Count (Nombre d'exemples), tout centile entre p0.0 et p100
`HeadRequests`	Nombre de demandes HTTP `HEAD` exécutées sur un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda. Unités : nombre Statistiques valides : somme
`ListRequests`	Le nombre de requêtes HTTP `GET` qui répertorient le contenu d'un compartiment Amazon S3. Cette métrique inclut les opérations `ListObjects` et `ListObjectsV2` . Unités : nombre Statistiques valides : somme
`4xxErrors`	Le nombre de demandes de code d'état d'erreur du client HTTP 4 xx adressées à un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda avec une valeur de 0 ou 1. La statistique Average (Moyenne) indique le taux d'erreur et la statistique Sum (Somme) indique le nombre de ce type d'erreur pendant chaque période. Unités : nombre Statistiques valides : Moyenne (rapports par demande), Somme (rapports par période), Min, Max, Exemple de comptage

Métrique	Description
`5xxErrors`	Le nombre de demandes de code d'état d'erreur du serveur HTTP 5 xx adressées à un compartiment Amazon S3 à l'aide d'un point d'accès Object Lambda avec une valeur de 0 ou 1. La statistique Average (Moyenne) indique le taux d'erreur et la statistique Sum (Somme) indique le nombre de ce type d'erreur pendant chaque période. Unités : nombre Statistiques valides : Moyenne (rapports par demande), Somme (rapports par période), Min, Max, Exemple de comptage
`ProxiedRequests`	Nombre de demandes HTTP adressées à un point d'accès Object Lambda qui renvoient la réponse d'API Amazon S3 standard. (Aucune fonction Lambda n'est configurée pour ces demandes.) Unités : nombre Statistiques valides : somme
`InvokedLambda`	Nombre de requêtes HTTP adressées à un objet S3 pour lesquelles une fonction Lambda a été appelée. Unités : nombre Statistiques valides : somme
`LambdaResponseRequests`	Nombre de requêtes `WriteGetObjectResponse` effectuées par la fonction Lambda. Cette métrique s'applique uniquement aux requêtes `GetObject` .
`LambdaResponse4xx`	Le nombre d'erreurs client HTTP 4 xx qui se produisent lors d'un appel `WriteGetObjectResponse` depuis une fonction Lambda. Cette métrique fournit les mêmes informations que `4xxErrors` , mais uniquement pour les appels à `WriteGetObjectResponse` .

Métrique	Description
`LambdaResponse5xx`	Nombre d'erreurs de serveur HTTP 5 xx qui se produisent lors d'un appel `WriteGetObjectResponse` depuis une fonction Lambda. Cette métrique fournit les mêmes informations que `5xxErrors`, mais uniquement pour les appels à `WriteGetObjectResponse`.

Les statistiques d'Amazon S3 sur Outposts dans CloudWatch

Pour une liste des métriques utilisées pour S3 sur les buckets Outposts, consultez. CloudWatch CloudWatch métriques

Dimensions d'Amazon S3 dans CloudWatch

Les dimensions suivantes sont utilisées pour filtrer les métriques Amazon S3.

Dimension	Description
`BucketName`	Cette dimension filtre les données que vous demandez dans le compartiment identifié uniquement.
`StorageType`	Cette dimension filtre les données que vous avez enregistrées dans un compartiment en fonction des types de stockage : • `StandardStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage `STANDARD`. • `IntelligentTieringAAStorage` – Nombre d'octets utilisés pour les objets du niveau d'accès Archive de la classe de stockage `INTELLIGENT_TIERING`. • `IntelligentTieringAIAStorage` – Nombre d'octets utilisés pour les objets du niveau d'accès Archive Instant de la classe de stockage `INTELLIGENT_TIERING`. • `IntelligentTieringDAAStorage` – Nombre d'octets utilisés pour les objets du niveau d'accès Deep Archive de la classe de stockage `INTELLIGENT_TIERING`.

Dimension	Description
	• `IntelligentTieringFAStorage` – Nombre d'octets utilisés pour les objets du niveau Accès fréquent de la classe de stockage `INTELLIGENT_TIERING` . • `IntelligentTieringIAStorage` – Nombre d'octets utilisés pour les objets du niveau Accès peu fréquent de la classe de stockage `INTELLIGENT_TIERING` . • `StandardIAStorage` — Le nombre d'octets utilisés pour les objets de la classe de stockage S3 Standard-Infrequent Access ()`STANDARD_IA` . • `StandardIASizeOverhead` – Nombre d'octets utilisés pour les objets d'une taille inférieure à 128 Ko dans la classe de stockage `STANDARD_IA` . • `IntAAObjectOverhead` – Pour chaque objet de la classe de stockage `INTELLIGENT_TIERING` au niveau d'accès Archive, S3 Glacier ajoute 32 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Glacier Flexible Retrieval vous est facturé pour ce stockage supplémentaire. • `IntAAS3ObjectOverhead` – Pour chaque objet de la classe de stockage `INTELLIGENT_TIERING` au niveau d'accès Archive, Amazon S3 utilise 8 Ko de stockage pour le nom de l'objet et les autres métadonnées. Le tarif S3 Standard vous est facturé pour ce stockage supplémentaire. • `IntDAAObjectOverhead` – Pour chaque objet de la classe de stockage `INTELLIGENT_TIERING` au niveau d'accès Deep Archive, S3 Glacier ajoute 32 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Glacier Deep Archive vous est facturé pour ce stockage supplémentaire. • `IntDAAS3ObjectOverhead` – Pour chaque objet de la classe de stockage `INTELLIGENT_TIERING` au niveau

Dimension	Description
	d'accès Deep Archive, Amazon S3 ajoute 8 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Standard vous est facturé pour ce stockage supplémentaire. • `OneZoneIAStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage S3 Unizone – Accès peu fréquent (`ONEZONE_IA`). • `OneZoneIASizeOverhead` – Nombre d'octets utilisés pour les objets d'une taille inférieure à 128 Ko dans la classe de stockage `ONEZONE_IA`. • `ReducedRedundancyStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage à redondance réduite (RRS). • `GlacierInstantRetrievalSizeOverhead` – Nombre d'octets utilisés pour les objets de taille inférieure à 128 Ko dans la classe de stockage S3 Glacier Instant Retrieval. • `GlacierInstantRetrievalStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage S3 Glacier Instant Retrieval. • `GlacierStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage S3 Glacier Flexible Retrieval. • `GlacierStagingStorage` – Nombre d'octets utilisés pour les parties des objets partitionnés avant la fin de l'exécution de la requête `CompleteMultipartUpload` sur des objets dans la classe de stockage S3 Glacier Flexible Retrieval. • `GlacierObjectOverhead` – Pour chaque objet archivé, S3 Glacier ajoute 32 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Glacier Flexible Retrieval vous est facturé pour ce stockage supplémentaire.

Dimension	Description
	• `GlacierS3ObjectOverhead` – Pour chaque objet archivé dans S3 Glacier Flexible Retrieval, Simple Storage Service (Amazon S3) utilise 8 Ko de stockage pour le nom de l'objet et d'autres métadonnées. Le tarif S3 Standard vous est facturé pour ce stockage supplémentaire. • `DeepArchiveStorage` – Nombre d'octets utilisés pour les objets dans la classe de stockage S3 Glacier Deep Archive. • `DeepArchiveObjectOverhead` – Pour chaque objet archivé, S3 Glacier ajoute 32 Ko de stockage pour l'index et les métadonnées associées. Ces données supplémentaires sont nécessaires pour identifier et restaurer l'objet. Le tarif S3 Glacier Deep Archive vous est facturé pour ce stockage supplémentaire. • `DeepArchiveS3ObjectOverhead` – Pour chaque objet archivé dans S3 Glacier Deep Archive, Simple Storage Service (Amazon S3) utilise 8 Ko de stockage pour le nom de l'objet et d'autres métadonnées. Le tarif S3 Standard vous est facturé pour ce stockage supplémentaire. • `DeepArchiveStagingStorage` – Nombre d'octets utilisés pour les parties des objets partitionnés avant la fin de l'exécution de la requête `CompleteMultipartUpload` sur des objets dans la classe de stockage S3 Glacier Deep Archive. • `ExpressOneZone` : nombre d’octets utilisés pour les objets dans la classe de stockage S3 Express One Zone.
`FilterId`	Cette dimension filtre les configurations de métriques que vous spécifiez pour les métriques de demande sur un compartim ent. Lorsque vous créez une configuration de métriques, vous spécifiez un ID de filtre (par exemple, un préfixe, une balise ou un point d'accès). Pour plus d'informations, voir Création d'une configuration de métriques.

Dimensions de réplication S3 dans CloudWatch

Les dimensions suivantes sont utilisées pour filtrer les métriques de réplication S3.

Dimension	Description
`SourceBucket`	Le nom du bucket à partir duquel les objets sont répliqués.
`DestinationBucket`	Le nom du bucket vers lequel les objets sont répliqués.
`RuleId`	Identifiant unique pour la règle qui a déclenché la mise à jour de cette métrique de réplication.

Dimensions de l'objectif de rangement S3 en CloudWatch

Pour obtenir la liste des dimensions utilisées pour filtrer les métriques de S3 Storage Lens CloudWatch, consultezDimensions.

Dimensions de la demande Lambda de l'objet S3 dans CloudWatch

Les dimensions suivantes sont utilisées pour filtrer les données à partir d'un point d'accès Object Lambda.

Dimension	Description
`AccessPointName`	Le nom du point d'accès pour lequel les demandes sont effectuées.
`DataSourceARN`	La source à partir de laquelle le point d'accès Object Lambda récupère les données. Si la demande invoque une fonction Lambda, cela fait référence au nom de ressource Lambda Amazon (ARN). Sinon, cela fait référence à l'ARN du point d'accès.

Accès aux CloudWatch métriques

Vous pouvez utiliser les procédures suivantes pour afficher les métriques de stockage d'Amazon S3. Pour obtenir les métriques Amazon S3 concernées, vous devez définir un horodatage de début et de fin. Pour les métriques portant sur une période de 24 heures, définissez la période sur

86 400 secondes, c'est-à-dire le nombre de secondes dans une journée. Vous devez également définir les dimensions `BucketName` et `StorageType`.

En utilisant le AWS CLI

Par exemple, si vous souhaitez utiliser le AWS CLI pour obtenir la moyenne de la taille d'un compartiment spécifique en octets, vous pouvez utiliser la commande suivante :

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
 --start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
 --unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=DOC-EXAMPLE-BUCKET
 Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Cet exemple produit le résultat suivant :

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:00:00Z",
            "Average": 1025328.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "BucketSizeBytes"
}
```

Utilisation de la console S3

Pour consulter les statistiques à l'aide de la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Metrics (Métriques).
3. Choisissez l'espace de nom S3.
4. (Facultatif) Pour afficher une métrique, saisissez son nom dans la zone de recherche.
5. (Facultatif) Pour filtrer par StorageTypedimension, entrez le nom de la classe de stockage dans le champ de recherche.

Pour consulter la liste des mesures valides enregistrées pour votre compte à Compte AWS l'aide du AWS CLI

- À partir d'une invite de commande, utilisez la commande suivante :

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Pour plus d'informations sur les autorisations requises pour accéder aux CloudWatch tableaux de bord, consultez les autorisations des tableaux CloudWatch de bord Amazon dans le guide de l' CloudWatch utilisateur Amazon.

CloudWatch configurations de métriques

Avec Amazon CloudWatch Request Metrics pour Amazon S3, vous pouvez recevoir des CloudWatch métriques d'une minute, configurer des CloudWatch alarmes et accéder à CloudWatch des tableaux de bord pour visualiser les near-real-time opérations et les performances de votre stockage Amazon S3. Pour les applications qui dépendent d'un stockage dans le Cloud, ces métriques vous permettent d'identifier rapidement les problèmes opérationnels et d'agir en conséquence. Lorsqu'elles sont activées, ces métriques d'1 minute sont disponibles par défaut au niveau du compartiment Amazon S3.

Si vous souhaitez obtenir les métriques de CloudWatch demande pour les objets d'un bucket, vous devez créer une configuration de métriques pour le bucket. Pour plus d’informations, consultez Création d'une configuration de CloudWatch métriques pour tous les objets de votre compartiment.

Vous pouvez également utiliser un préfixe partagé, des balises d'objet ou un point d'accès pour définir un filtre pour les métriques collectées. Cette méthode de définition de filtre vous permet de faire correspondre les filtres des métriques à des applications métier, des flux de travail ou des organisations internes spécifiques. Pour plus d’informations, consultez Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès. Pour plus d'informations sur les CloudWatch métriques disponibles et les différences entre les métriques de stockage et les métriques de demande, consultezSurveillance des métriques avec Amazon CloudWatch.

Gardez ces conseils à l'esprit lorsque vous utilisez les configurations de métriques :

- Vous pouvez avoir 1 000 configurations de métriques maximum par compartiment.
- Vous pouvez utiliser des filtres pour choisir les objets d'un compartiment à inclure dans les configurations de métriques. Vous pouvez filtrer sur un préfixe partagé, une balise d'objet ou

un point d'accès pour aligner les filtres des métriques sur des applications commerciales, des flux de travail ou des organisations internes spécifiques. Pour demander des métriques pour le compartiment entier, créez une configuration de métriques sans filtre.

- Les configurations de métriques ne sont nécessaires que pour activer les métriques de demandes. Les métriques quotidiennes de stockage au niveau des compartiments sont toujours activées et sont fournies sans frais supplémentaires. Actuellement, il n'est pas possible d'obtenir des métriques de stockage quotidiennes pour un sous-ensemble d'objets filtré.
- Chaque configuration de métriques active l'ensemble complet de métriques de demandes disponibles. Les métriques spécifiques à une opération (telles que `PostRequests`) ne sont indiquées que s'il existe des demandes de ce type pour votre compartiment ou votre filtre.
- Les métriques de demandes sont indiquées pour toutes les opérations au niveau des objets. Elles sont également signalées pour les opérations qui répertorient le contenu du compartiment, comme GET Bucket (List Objects), GET Bucket Object Versions et List Multipart Uploads. En revanche, elles ne sont pas signalées pour les autres opérations sur les compartiments.
- Les métriques de demandes prennent en charge le filtrage par préfixe, balise d'objet ou point d'accès, contrairement aux métriques de stockage.

Livraison des CloudWatch indicateurs les plus efficaces

CloudWatch les indicateurs sont fournis dans la mesure du possible. La plupart des demandes relatives à un objet Amazon S3 comportant des métriques de demande aboutissent à l'envoi d'un point de données à CloudWatch.

L'exhaustivité et la ponctualité des métriques ne sont pas garanties. Le point de données d'une demande particulière doit être retourné avec un horodatage ultérieur au moment du traitement de la demande. Le point de données peut être retardé d'une minute avant d'être disponible CloudWatch, ou il se peut qu'il ne soit pas livré du tout. CloudWatch les métriques de demande vous donnent une idée de la nature du trafic par rapport à votre compartiment en temps quasi réel. L'objectif n'est pas de comptabiliser toutes les demandes.

En conséquence de la nature du « meilleur effort » de cette fonction, les rapports disponibles sur le Tableau de bord Gestion de la facturation et des coûts peuvent inclure une ou plusieurs demandes d'accès qui ne figurent pas dans les métriques du compartiment.

Pour plus d'informations sur l'utilisation des CloudWatch métriques dans Amazon S3, consultez les rubriques suivantes.

Rubriques

- Création d'une configuration de CloudWatch métriques pour tous les objets de votre compartiment
- Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès
- Suppression d’un filtre de métriques

Création d'une configuration de CloudWatch métriques pour tous les objets de votre compartiment

Lorsque vous configurez les métriques de demande, vous pouvez créer une configuration de CloudWatch métriques pour tous les objets de votre bucket, ou vous pouvez filtrer par préfixe, balise d'objet ou point d'accès. Les procédures décrites dans cette rubrique vous montrent comment créer une configuration pour tous les objets de votre compartiment. Pour créer une configuration qui filtre par balise, préfixe d'objet ou point d'accès, consultez Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès.

Il existe trois types de CloudWatch métriques Amazon pour Amazon S3 : les métriques de stockage, les métriques de demande et les métriques de réplication. Les métriques du stockage sont fournies une fois par jour à tous les clients sans frais supplémentaires. Ces métriques de demande sont disponibles à des intervalles d'une minute après une latence pour le traitement. Les métriques relatives aux demandes sont facturées au CloudWatch tarif standard. Vous devez activer les métriques de demande en les configurant dans la console ou en utilisant l'API Amazon S3. Les métriques de réplication S3 fournissent des métriques détaillées pour les règles de réplication dans votre configuration de réplication. Grâce aux métriques de réplication, vous pouvez suivre la minute-by-minute progression en suivant les octets en attente, les opérations en attente, les opérations dont la réplication a échoué et la latence de réplication.

Pour plus d'informations sur CloudWatch les métriques pour Amazon S3, consultezSurveillance des métriques avec Amazon CloudWatch.

Vous pouvez ajouter des configurations de métriques à un compartiment à l'aide de la console Amazon S3, de l' AWS Command Line Interface (AWS CLI) ou de l'API REST Amazon S3.

Utiliser la console S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiment, choisissez le nom du compartiment qui contient les objets pour lesquels vous souhaitez les métriques des demandes.

3. Sélectionnez l'onglet Métriques.
4. Sous Bucket metrics (Métriques de compartiment), choisissez View additional charts (Afficher des graphiques supplémentaires).
5. Choisissez l'onglet Request metrics (Métriques de demande).
6. Choisissez Create filter (Créer un filtre).
7. Dans la zone Filter name (Nom du filtre), saisissez un nom de filtre.

   Les noms peuvent contenir uniquement des lettres, des chiffres, des points, des tirets et des traits de soulignement. Nous vous recommandons d'utiliser le nom `EntireBucket` pour un filtre qui s'applique à tous les objets.
8. Sous Portée de filtre, choisissez Ce filtre s'applique à tous les objets du compartiment.

   Vous pouvez également définir un filtre pour collecter les métriques relatives à un sous-ensemble d'objets stockés dans le compartiment et générer le rapport correspondant. Pour plus d'informations, consultez Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès.
9. Choisissez Enregistrer les modifications.
10. Sous l'onglet Métriques de demande, sous Filtres, choisissez le filtre que vous venez de créer.

    Après environ 15 minutes, CloudWatch commence à suivre ces métriques de demande. Vous pouvez les afficher dans l'onglet Request metrics (Métriques de demande). Vous pouvez consulter les graphiques des métriques sur Amazon S3 ou sur CloudWatch la console. Les métriques relatives aux demandes sont facturées au CloudWatch tarif standard. Pour plus d'informations, consultez les CloudWatch tarifs Amazon.

Utilisation de l'API REST

Vous pouvez également ajouter des configurations de métriques par programmation avec l'API REST Amazon S3. Pour en savoir plus sur l'ajout et l'utilisation de configurations de métriques, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service :

- Configuration de métriques PUT Bucket
- Configuration de métriques GET Bucket
- Configuration de métriques List Bucket
- Configuration de métriques DELETE Bucket

À l'aide du AWS CLI

1. Installez et configurez le AWS CLI. Pour obtenir des instructions, consultez la section Installation, mise à jour et désinstallation de la AWS CLI du Guide de l'utilisateur de la AWS Command Line Interface .
2. Ouvrez un terminal.
3. Exécutez la commande suivante pour ajouter une configuration de métriques :

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.us-
west-2.amazonaws.com --bucket bucket-name --id metrics-config-id --metrics-
configuration '{"Id":"metrics-config-id"}'
```

Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès

Il existe trois types de CloudWatch métriques Amazon pour Amazon S3 : les métriques de stockage, les métriques de demande et les métriques de réplication. Les métriques du stockage sont fournies une fois par jour à tous les clients sans frais supplémentaires. Ces métriques de demande sont disponibles à des intervalles d'une minute après une latence pour le traitement. Les métriques relatives aux demandes sont facturées au CloudWatch tarif standard. Vous devez activer les métriques de demande en les configurant dans la console ou en utilisant l'API Amazon S3. Les métriques de réplication S3 fournissent des métriques détaillées pour les règles de réplication dans votre configuration de réplication. Grâce aux métriques de réplication, vous pouvez suivre la minute-by-minute progression en suivant les octets en attente, les opérations en attente, les opérations dont la réplication a échoué et la latence de réplication.

Pour plus d'informations sur CloudWatch les métriques pour Amazon S3, consultezSurveillance des métriques avec Amazon CloudWatch.

Lorsque vous configurez CloudWatch les métriques, vous pouvez créer un filtre pour tous les objets de votre bucket, ou vous pouvez filtrer la configuration en groupes d'objets connexes au sein d'un même bucket. Vous pouvez filtrer les objets d'un compartiment pour l'inclusion dans une configuration de métriques sur la base de l'un ou de plusieurs des types de filtre suivants :

- Préfixe de nom de clé d'objet – Même si le modèle de données Amazon S3 est une structure plane, vous pouvez induire une hiérarchie au moyen d'un préfixe. La console Amazon S3 prend en charge ces préfixes avec le concept de dossiers. Si vous filtrez par préfixe, les objets ayant le même

préfixe seront inclus dans la configuration des métriques. Pour en savoir plus sur les préfixes, consultez Organisation des objets à l'aide de préfixes.

- Balise – Vous pouvez ajouter des balises, à savoir des paires de noms clés-valeurs, aux objets. Les balises vous permettent de trouver et de classer facilement les objets. Ces balises peuvent également être utilisées comme filtres pour les configurations de métriques. Pour en savoir plus sur les balises d'objet, consultez Catégorisation de votre stockage à l'aide de balises.
- Point d'accès – Les points d'accès S3 sont nommés points de terminaison réseaux qui sont attachés à des compartiments et simplifient la gestion de l'accès aux données à l'échelle pour les jeux de données partagés dans S3. Lorsque vous créez un filtre de point d'accès, Amazon S3 inclut les demandes au point d'accès que vous indiquez dans la configuration des métriques. Pour plus d'informations, consultez Surveillance et journalisation des points d'accès.

> **Note**
>
> Lorsque vous créez une configuration de métriques qui filtre par point d'accès, vous devez utiliser Amazon Resource Name (ARN) du point d'accès, et non l'alias du point d'accès. Veillez à utiliser l'ARN pour le point d'accès lui-même, et non l'ARN pour un objet particulier. Pour plus d'informations sur les ARN des points d'accès, consultez Utilisation des points d'accès.

Si vous spécifiez un filtre, seules les demandes qui agissent sur des objets uniques peuvent correspondre au filtre et être incluses dans les métriques indiquées. Les requêtes telles que DeleteObjectset les `ListObjects` requêtes ne renvoient aucune métrique pour les configurations avec filtres.

Pour demander un filtrage plus complexe, choisissez deux ou plusieurs éléments. Seuls les objets possédant tous ces éléments sont inclus dans la configuration des métriques. Si vous ne définissez pas de filtres, tous les objets du compartiment sont inclus dans la configuration des métriques.

Utiliser la console S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiment, choisissez le nom du compartiment qui contient les objets pour lesquels vous souhaitez les métriques des demandes.
3. Choisissez l'onglet Métriques.

4. Sous Bucket metrics (Métriques de compartiment), choisissez View additional charts (Afficher des graphiques supplémentaires).
5. Choisissez l'onglet Request metrics (Métriques de demande).
6. Choisissez Create filter (Créer un filtre).
7. Dans la zone Filter name (Nom du filtre), saisissez un nom de filtre.

   Les noms peuvent contenir uniquement des lettres, des chiffres, des points, des tirets et des traits de soulignement.
8. Sous Portée du filtre, choisissez Limiter la portée de ce filtre à l'aide d'un préfixe, de balises d'objet et d'un point d'accès S3, ou d'une combinaison des trois.
9. Sous Type de filtre, choisissez au moins un type de filtre : Préfixe, Balises de ouPoint d'accès.
10. Pour définir un filtre de préfixe et limiter la porté du filtre à un seul chemin, saisissez un préfixe dans la zone Préfixe.
11. Pour définir un filtre de balises d'objet, sous Balises d'objet, choisissez Ajouter une balise, puis saisissez une balise Clé et Valeur.
12. Pour définir un filtre de point d'accès, dans le champ Point d'accès S3, saisissez l'ARN du point d'accès ou choisissez Parcourir S3 pour parvenir au point d'accès.

    > **Important**
    >
    > Vous ne pouvez pas saisir l'alias d'un point d'accès. Vous devez saisir l'ARN du point d'accès lui-même, et non l'ARN d'un objet spécifique.
13. Sélectionnez Enregistrer les modifications.

    Amazon S3 crée un filtre qui utilise le préfixe, les balises ou le point d'accès que vous avez indiqués.
14. Sous l'onglet Métriques de demande, sous Filtres, choisissez le filtre que vous venez de créer.

    Vous avez maintenant créé un filtre qui limite la portée des métriques des demandes par préfixe, balises d'objet ou point d'accès. Environ 15 minutes après le CloudWatch début du suivi de ces métriques de demande, vous pouvez consulter les graphiques des métriques sur Amazon S3 et sur CloudWatch les consoles. Les métriques relatives aux demandes sont facturées au CloudWatch tarif standard. Pour plus d'informations, consultez les CloudWatch tarifs Amazon.

Vous pouvez également configurer les métriques des demandes au niveau d'un compartiment. Pour plus d'informations, veuillez consulter Création d'une configuration de CloudWatch métriques pour tous les objets de votre compartiment.

À l'aide du AWS CLI

1. Installez et configurez le AWS CLI. Pour obtenir des instructions, consultez la section Installation, mise à jour et désinstallation de la AWS CLI du Guide de l'utilisateur de la AWS Command Line Interface .
2. Ouvrez un terminal.
3. Pour ajouter une configuration de métriques, exécutez l'une des commandes suivantes :

   Example : pour filtrer par préfixe

   ```
   aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
   id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
   {"Prefix":"prefix1"}} '
   ```

   Example : pour filtrer par balises

   ```
   aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
   id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
   {"Tag": {"Key": "string", "Value": "string"}} '
   ```

   Example : pour filtrer par point d'accès

   ```
   aws s3api put-bucket-metrics-configuration --bucket DOC-EXAMPLE-BUCKET1 --
   id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
   {"AccessPointArn":"arn:aws:s3:Region:account-id:accesspoint/access-point-name"}} '
   ```

   Example : pour filtrer par préfixe, balises et point d'accès

   ```
   aws s3api put-bucket-metrics-configuration --endpoint https://
   s3.Region.amazonaws.com --bucket DOC-EXAMPLE-BUCKET1 --id metrics-config-id --
   metrics-configuration '
   {
       "Id": "metrics-config-id",
       "Filter": {
   ```

```
        "And": {
            "Prefix": "string",
            "Tags": [
                {
                    "Key": "string",
                    "Value": "string"
                }
            ],
            "AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-
point-name"
        }
    }
}'
```

Utilisation de l'API REST

Vous pouvez également ajouter des configurations de métriques par programmation avec l'API REST Amazon S3. Pour en savoir plus sur l'ajout et l'utilisation de configurations de métriques, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service :

- Configuration de métriques PUT Bucket
- Configuration de métriques GET Bucket
- Configuration de métriques List Bucket
- Configuration de métriques DELETE Bucket

Suppression d’un filtre de métriques

Vous pouvez supprimer un filtre de statistiques de CloudWatch demande Amazon si vous n'en avez plus besoin. Lorsque vous supprimez un filtre, les métriques des demandes qui utilisent ce filtre spécifiquene vous sont plus facturées. Toutefois, vous continuerez à être facturé pour toute autre configuration de filtre existante.

Lorsque vous supprimez un filtre, vous ne pouvez plus l'utiliser pour les métriques des demandes. La suppression d'un filtre ne peut pas être annulée.

Pour plus d'informations sur la création d'un filtre de métriques des demandes, consultez les rubriques suivantes :

- Création d'une configuration de CloudWatch métriques pour tous les objets de votre compartiment

- Création d'une configuration de métriques qui filtre par préfixe, balise d'objet ou point d'accès

Utilisation de la console S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom de votre compartiment.
3. Sélectionnez l'onglet Métriques.
4. Sous Bucket metrics (Métriques de compartiment), choisissez View additional charts (Afficher des graphiques supplémentaires).
5. Choisissez l'onglet Request metrics (Métriques de demande).
6. Choisissez Manage filters (Gérer les filtres).
7. Choisissez votre filtre.

   > **Important**
   >
   > La suppression d'un filtre ne peut pas être annulée.

8. Sélectionnez Delete.

   Amazon S3 supprime votre filtre.

Utilisation de l'API REST

Vous pouvez également ajouter des configurations de métriques par programmation avec l'API REST Amazon S3. Pour en savoir plus sur l'ajout et l'utilisation de configurations de métriques, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service :

- Configuration de métriques PUT Bucket
- Configuration de métriques GET Bucket
- Configuration de métriques List Bucket
- Configuration de métriques DELETE Bucket

Notifications d'événements Amazon S3

Vous pouvez utiliser la fonctionnalité Notifications d'événements Amazon S3 pour recevoir des notifications lorsque certains événements se produisent dans votre compartiment S3. Pour activer les notifications, ajoutez une configuration de notification qui identifie les événements que Amazon S3 doit publier. Assurez-vous que les destinations auxquelles Amazon S3 doit envoyer les notifications sont également identifiées. Vous stockez cette configuration dans la sous-ressource notification associée au compartiment. Pour plus d'informations, consultez Options de configuration des compartiments. Amazon S3 fournit une API qui vous permet de gérer cette sous-ressource.

> **Important**
>
> Les notifications d'événement Amazon S3 sont conçues pour être délivrées au moins une fois. Les notifications d'événements parviennent généralement à destination en quelques secondes, mais cela peut aussi prendre une minute ou plus.

Présentation des notifications d'événements Amazon S3.

Actuellement, Amazon S3 peut publier des notifications pour les événements suivants :

- Événements créés par un nouvel objet
- Événements de suppression d'objets
- Événements de restauration d'objet
- Un événement de perte d'un objet Reduced Redundancy Storage (RRS)
- Événements de réplication
- Événements d'expiration du cycle de vie S3
- Événements de transition du cycle de vie S3
- Événements d'archivage automatique S3 Intelligent-Tiering
- Événements de balisage d'objets
- Événements PUT de liste ACL de l'objet

Pour obtenir des descriptions complètes de tous les types d'événement pris en charge, consultez Types d'événements pris en charge pour SQS, SNS et Lambda.

Amazon S3 peut envoyer des messages de notification d'événement vers les types de destination suivants : Vous devez spécifier l'Amazon Resource Name (ARN) de ces destinations dans la configuration des notifications.

- Rubriques Amazon Simple Notification Service (Amazon SNS)
- Files d'attente Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda fonction
- Amazon EventBridge

Pour plus d’informations, consultez Destinations d'événements prises en charge.

> **Note**
>
> Les files d'attente Amazon Simple Queue Service FIFO (premier entré, premier sorti) ne sont pas prises en charge en tant que destination des notifications d'événements Amazon S3. Pour envoyer une notification concernant un événement Amazon S3 à une file d'attente FIFO Amazon SQS, vous pouvez utiliser Amazon. EventBridge Pour plus d’informations, consultez Activation d'Amazon EventBridge.

> **Warning**
>
> Lorsque votre notification écrit dans le compartiment qui déclenche la notification, cela peut provoquer une boucle d'exécution. Par exemple, si le compartiment déclenche une fonction Lambda chaque fois qu'un objet est chargé et que la fonction charge un objet dans le compartiment, elle se déclenche elle-même indirectement. Afin d'éviter cela, utilisez deux compartiments ou configurez le déclencheur pour qu'il s'applique uniquement à un préfixe utilisé pour les objets entrants.
>
> Pour plus d'informations et un exemple d'utilisation des notifications Amazon S3 avec AWS Lambda, consultez la section Utilisation AWS Lambda avec Amazon S3 dans le guide du AWS Lambda développeur.

Pour plus d'informations sur le nombre de configurations de notifications d'événements que vous pouvez créer par compartiment, consultez Quotas de service Amazon S3 dans la Référence générale d'AWS .

Pour plus d'informations sur les notifications d'événements, consultez les sections suivantes.

Rubriques

Types de notification d'événements et destinations associées

Amazon S3 prend en charge plusieurs types et destinations de notifications d'événements. Vous pouvez spécifier le type d'événement et la destination lors de la configuration de vos notifications. Une seule destination peut être spécifiée pour chaque notification d'événement. Les notifications d'événements Amazon S3 envoient une entrée d'événement pour chaque message de notification.

Rubriques

Destinations d'événements prises en charge

Amazon S3 peut envoyer des messages de notification d'événement vers les types de destination suivants :

- Rubriques Amazon Simple Notification Service (Amazon SNS)
- Files d'attente Amazon Simple Queue Service (Amazon SQS)
- AWS Lambda
- Amazon EventBridge

Cependant, un seul type de destination peut être spécifié pour chaque notification d'événement.

> **Note**
>
> Vous devez autoriser Amazon S3 à publier des messages dans une rubrique Amazon SNS ou une file d'attente Amazon SQS. Vous devez également autoriser Amazon S3 à appeler une AWS Lambda fonction en votre nom. Pour obtenir des instructions sur la façon d'accorder ces autorisations, consultez Octroi d'autorisations pour la publication de messages de notification d'événement vers une destination.

Rubrique Amazon SNS

Amazon SNS est un service de messagerie de type Push flexible et totalement géré. Vous pouvez utiliser ce service pour envoyer des messages en mode Push vers des appareils mobiles ou des services distribués. Grâce à SNS, vous pouvez publier un message, puis le diffuser à une ou plusieurs reprises. Actuellement, SNS standard n'est autorisé qu'en tant que destination de notification d'événement S3, tandis que les rubriques FIFO SNS ne sont pas autorisées.

Amazon SNS coordonne et gère l'envoi et la diffusion de messages aux points de terminaison ou aux clients abonnés. Vous pouvez utiliser la console Amazon SNS pour créer la rubrique Amazon SNS qui recevra vos notifications.

Le sujet doit se trouver dans le même Région AWS que celui de votre compartiment Amazon S3. Pour obtenir des instructions sur comment créer une rubrique Amazon SNS, veuillez consulter la section Démarrer avec Amazon SNS du Manuel du développeur Amazon Simple Notification Service, ainsi que les Questions fréquentes (FAQ) sur Amazon SNS.

Pour pouvoir utiliser la rubrique Amazon SNS que vous avez créé en tant que destination d'une notification d'événements, vous devez disposer des éléments suivants :

- Amazon Resource Name (ARN) de la rubrique Amazon SNS
- Un abonnement à la rubrique Amazon SNS valide. Grâce à cela, les abonnés à la rubrique reçoivent une notification lorsqu'un message est publié sur votre rubrique Amazon SNS.

File d’attente Amazon SQS

Amazon SQS offre des files d'attente hébergées, évolutives et fiables dédiées au stockage des messages pendant leur circulation entre les ordinateurs. Amazon SQS vous permet de transmettre n'importe quel volume de données, sans nécessiter que d'autres services soient toujours disponibles.

Vous pouvez utiliser la console Amazon SQS pour créer une file d'attente Amazon SQS qui recevra vos notifications.

La file d'attente Amazon SQS doit être Région AWS identique à celle de votre compartiment Amazon S3. Pour obtenir des instructions sur comment créer une file d'attente Amazon SQS, veuillez consulter les sections Qu'est-ce qu'Amazon Simple Queue Service et Mise en route avec Amazon SQS du Manuel du développeur Amazon Simple Queue Service.

Pour pouvoir utiliser la file d'attente Amazon SQS en tant que destination d'une notification d'événements, vous devez disposer des éléments suivants :

- Amazon Resource Name (ARN) de la file d'attente Amazon SQS

> **Note**
>
> Les files d'attente Amazon Simple Queue Service FIFO (premier entré, premier sorti) ne sont pas prises en charge en tant que destination des notifications d'événements Amazon S3. Pour envoyer une notification concernant un événement Amazon S3 à une file d'attente FIFO Amazon SQS, vous pouvez utiliser Amazon. EventBridge Pour plus d'informations, consultez Activation d'Amazon EventBridge.

Fonction Lambda

Vous pouvez l'utiliser AWS Lambda pour étendre d'autres AWS services avec une logique personnalisée ou créer votre propre backend qui fonctionne en termes d' AWS échelle, de performances et de sécurité. Lambda vous permet de créer des applications discrètes et axées sur les événements qui ne s'exécutent que lorsque vous en avez besoin. Vous pouvez également l'utiliser pour faire évoluer ces applications automatiquement de quelques requêtes par jour à des milliers de secondes.

Lambda peut exécuter du code personnalisé en réponse aux événements de compartiment Amazon S3 : vous chargez votre code personnalisé dans Lambda et créez ce qu'on appelle une fonction Lambda. Lorsqu'Amazon S3 détecte un événement d'un type spécifique, il peut le publier AWS Lambda et appeler votre fonction dans Lambda. En réponse, Lambda exécute votre fonction. Un type d'événement qu'il peut détecter, par exemple, est un événement créé par un objet.

Vous pouvez utiliser la AWS Lambda console pour créer une fonction Lambda qui utilise l' AWS infrastructure pour exécuter le code en votre nom. La fonction Lambda doit être située dans la même

Région que votre compartiment S3. Vous devez également connaître le nom ou l'ARN d'une fonction Lambda pour la configurer en tant que destination de notification d'événements.

> **Warning**
>
> Lorsque votre notification écrit dans le compartiment qui déclenche la notification, cela peut provoquer une boucle d'exécution. Par exemple, si le compartiment déclenche une fonction Lambda chaque fois qu'un objet est chargé et que la fonction charge un objet dans le compartiment, elle se déclenche elle-même indirectement. Afin d'éviter cela, utilisez deux compartiments ou configurez le déclencheur pour qu'il s'applique uniquement à un préfixe utilisé pour les objets entrants.
>
> Pour plus d'informations et un exemple d'utilisation des notifications Amazon S3 avec AWS Lambda, consultez la section Utilisation AWS Lambda avec Amazon S3 dans le guide du AWS Lambda développeur.

Amazon EventBridge

Amazon EventBridge est un bus d'événements sans serveur qui reçoit les événements des AWS services. Vous pouvez définir des règles pour faire correspondre les événements et les transmettre à des cibles, par exemple un service AWS ou un point de terminaison HTTP. Pour plus d'informations, consultez le contenu EventBridge du guide de EventBridge l'utilisateur Amazon.

Contrairement aux autres destinations, vous pouvez activer ou désactiver les événements à diffuser EventBridge pour un bucket. Si vous activez la livraison, tous les événements sont envoyés à EventBridge. En outre, vous pouvez utiliser des EventBridge règles pour acheminer les événements vers des cibles supplémentaires.

Types d'événements pris en charge pour SQS, SNS et Lambda

Amazon S3 peut publier les types d'événements ci-après. Vous devez spécifier ces types d'événements dans la configuration des notifications.

Types d'événements	Description
s3 : TestEvent	Lorsqu'une notification est activée, Amazon S3 publie une notification de test. Cela permet de vérifier que la rubrique existe et que le propriétaire du compartiment est autorisé à publier la rubrique spécifiée.

Types d'événements	Description
	Si l'activation de la notification échoue, vous ne recevez pas de notification de test.
s3 ObjectCreated : * s3 ::Put ObjectCreated s3 ::Publier ObjectCreated s3 ::Copier ObjectCreated s3 ObjectCreated : CompleteMultipartUpload	Les opérations API Amazon S3 telles que PUT, POST et COPY permettent de créer des objets. Avec ces types d'événements, vous pouvez activer les notifications lorsqu'un objet est créé à l'aide d'une opération API spécifique. Vous pouvez également utiliser le type d'événement `s3:ObjectCreated:*` pour demander une notification quelle que soit l'API utilisée pour créer un objet. `s3:ObjectCreated:CompleteMultipartUpload` inclut des objets créés à l'aide UploadPartCopyd'opérations de copie.

Types d'événements	Description
s3 ObjectRemoved : * s3 ::Supprimer ObjectRemoved s3 ObjectRemoved : DeleteMarkerCreated	En utilisant les types d'ObjectRemovedévénements, vous pouvez activer la notification lorsqu'un objet ou un lot d'objets est retiré d'un compartiment. Vous pouvez utiliser le type d'événement `s3:ObjectRemoved:Delete` pour demander l'envoi de notifications lorsqu'un objet est supprimé ou lorsqu'un objet avec version est supprimé de manière définitive. Vous pouvez également utiliser `s3:ObjectRemoved:DeleteMarkerCreated`, afin de demander l'envoi d'une notification lorsqu'un marqueur de suppression est créé pour un objet avec version. Pour savoir comment supprimer des objets versionnés, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions. Vous pouvez aussi utiliser un caractère générique `s3:ObjectRemoved:*`, afin de demander l'envoi d'une notification chaque fois qu'un objet est supprimé. Ces notifications d'événements ne vous alertent pas lors de suppressions automatiques dans le cadre de configurations de cycle de vie ou lorsque des opérations échouent.

Types d'événements	Description
s3 ObjectRestore : * s3 ::Publier ObjectRestore s3 : Terminé ObjectRestore s3 ::Supprimer ObjectRestore	En utilisant les types d'ObjectRestoreévénements, vous pouvez recevoir des notifications concernant le lancement et la fin d'un événement lors de la restauration d'objets issus de la classe de stockage S3 Glacier Flexible Retrieval , de la classe de stockage S3 Glacier Deep Archive, du niveau S3 Intelligent-Tiering Archive Access et du niveau S3 Intelligent-Tiering Deep Archive Access. Vous pouvez également recevoir des notifications lorsque la copie restaurée d'un objet expire. Le type d'événement `s3:ObjectRestore:Post` vous avertit du début d'une restauration d'objet. Le type d'événement `s3:ObjectRestore:Completed` vous avertit de la fin de la restauration. Le type d'événement `s3:ObjectRestore:Delete` vous avertit lorsque la copie temporaire d'un objet restauré expire.
s3 : ReducedRedundancyLostObject	Vous recevez cette notification lorsqu'Amazon S3 détecte qu'un objet de la classe de stockage RRS a été perdu.

Types d'événements	Description
s3:Replication:* S3 : réplication : Operation FailedReplication S3 : réplication : Operation MissedThreshold S3 : réplication : Operation ReplicatedAfterThreshold S3 : réplication : Operation NotTracked	L'utilisation des types d'événements de réplication vous permet de recevoir des notifications pour les configurations de réplication sur lesquelles des métriques de réplication S3 ou Contrôle du temps de réplication S3 (S3 RTC) sont activées. Vous pouvez surveiller la minute-by-minute progression des événements de réplication en suivant les octets en attente, les opérations en attente et la latence de réplication. Pour plus d'informations sur les métriques de réplication, consultez Surveillance de la progression avec des métriques de réplication et des notifications d'événements S3. Le type d'événement `s3:Replication:OperationFailedReplication` vous avertit lorsqu'un objet éligible à la réplication n'a pas pu être répliqué. Le type d'événement `s3:Replication:OperationMissedThreshold` vous avertit lorsqu'un objet éligible à la réplication dépasse le seuil de 15 minutes pour la réplication. Le type d'événement `s3:Replication:OperationReplicatedAfterThreshold` vous avertit lorsqu'un objet éligible à la réplication qui utilise S3 Replication Time Control est répliqué après le seuil de 15 minutes. Le type d'événement `s3:Replication:OperationNotTracked` vous avertit lorsqu'un objet éligible à la réplication qui utilise S3 Replication Time Control n'est plus suivi par les mesures de réplication.

Types d'événements	Description
s3 LifecycleExpiration : * s3 ::Supprimer LifecycleExpiration s3 LifecycleExpiration : DeleteMarkerCreated	En utilisant les types d'LifecycleExpirationévénements, vous pouvez recevoir une notification lorsqu'Amazon S3 supprime un objet en fonction de votre configuration S3 Lifecycle. Le type d'événement `s3:LifecycleExpiration:Delete` vous avertit lorsqu'un objet dans un compartiment non versionné est supprimé. Il vous avertit également lorsqu'une version d'objet est définitivement supprimée par une configuration de cycle de vie S3. Le type d'événement `s3:LifecycleExpiration:DeleteMarkerCreated` vous avertit lorsque le cycle de vie S3 crée un marqueur de suppression lors de la suppression d'une version actuelle d'un objet dans un compartiment versionné.
s3 : LifecycleTransition	Vous recevez cette notification d'événement lorsqu'un objet est transféré vers une autre classe de stockage Amazon S3 par une configuration de cycle de vie S3.
s3 : IntelligentTiering	Vous recevez cette notification d'événement afin d'être informé lorsqu'un objet de la classe de stockage S3 Intelligent-Tiering est passé au niveau d'accès Archive ou au niveau d'accès Deep Archive.
s3 ObjectTagging : * s3 ::Put ObjectTagging s3 ::Supprimer ObjectTagging	En utilisant les types d'ObjectTaggingévénements, vous pouvez activer la notification lorsqu'une balise d'objet est ajoutée ou supprimée d'un objet. Le type d'événement `s3:ObjectTagging:Put` vous avertit lorsqu'une étiquette est PUT sur un objet ou qu'une étiquette existante est mise à jour. Le type d'événement `s3:ObjectTagging:Delete` vous avertit lorsqu'une étiquette est supprimée d'un objet.

Types d'événements	Description
s3 ::Put ObjectAcl	Vous recevez cette notification d'événement lorsqu'une liste ACL est PUT sur un objet ou lorsqu'une liste ACL existante est modifiée. Un événement n'est pas généré lorsqu'une demande n'entraîne aucune modification de la liste ACL d'un objet.

Types d'événements pris en charge pour Amazon EventBridge

Pour obtenir la liste des types d'événements qu'Amazon S3 enverra à Amazon EventBridge, consultezEn utilisant EventBridge.

Organisation d'événements et doublons d'événements

Les notifications d'événements Amazon S3 sont conçues pour envoyer des notifications au moins une fois, mais il n'est pas garanti qu'elles arrivent dans le même ordre que celui dans lequel les événements se sont produits. Dans de rares cas, le mécanisme de nouvelle tentative d'Amazon S3 peut provoquer des notifications d'événement S3 dupliquées pour le même événement d'objet. Pour en savoir plus sur la gestion des événements dupliqués ou en rupture de commande, consultez Gérer l'ordre des événements et les événements dupliqués avec les notifications d'événements Amazon S3 sur le blog AWS de stockage.

Utilisation d'Amazon SQS, Amazon SNS et Lambda

L'activation des notifications s'effectue au niveau du compartiment. Vous stockez les informations de configuration de la notification dans la sous-ressource notification qui est associée à un compartiment. Après avoir créé ou modifié la configuration de notification du compartiment, il faut généralement attendre 5 minutes pour que les modifications prennent effet. Lorsque la notification est activée pour la première fois, un événement `s3:TestEvent` se produit. Suivez l'une des méthodes suivantes pour gérer la configuration des notifications :

- Via la console Amazon S3 – Vous pouvez utilisez l'interface de la console pour définir une configuration de notifications pour un compartiment sans écrire de code. Pour plus d'informations, consultez Activation et configuration des notifications d'événements à l'aide de la console Amazon S3.
- Utilisation des AWS SDK par programmation : en interne, la console et les SDK appellent l'API REST Amazon S3 pour gérer les sous-ressources de notification associées au compartiment.

Pour des exemples de configuration des notifications qui utilisent le kit AWS SDK, consultez Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS).

> **Note**
>
> Vous pouvez aussi appeler l'API REST Amazon S3 directement depuis votre code. Toutefois, cette méthode peut s'avérer fastidieuse, car vous devez écrire du code pour authentifier vos demandes.

Quelle que soit la méthode que vous utilisez, Amazon S3 stocke la configuration des notifications au format XML dans la sous-ressource notification qui est associée au compartiment. Pour plus d'informations sur les sous-ressources de compartiment, consultez Options de configuration des compartiments.

Rubriques

- Octroi d'autorisations pour la publication de messages de notification d'événement vers une destination
- Activation et configuration des notifications d'événements à l'aide de la console Amazon S3
- Configuration des notifications d'événements par programmation
- Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS)
- Configuration des notifications d'événement à l'aide du filtrage des noms de clé d'objet
- Structure des messages d'événements

Octroi d'autorisations pour la publication de messages de notification d'événement vers une destination

Vous devez accorder au principal Amazon S3 les autorisations nécessaires pour appeler l'API appropriée afin de publier des messages dans une rubrique SNS, une file d'attente SQS ou une fonction Lambda. Amazon S3 peut ainsi publier des messages de notification d'événement vers une destination.

Pour résoudre les problèmes liés à la publication de messages de notification d'événements vers une destination, consultez Troubleshoot to publish Amazon S3 event notifications to an Amazon Simple

Notification Service topic (Résoudre les problèmes liés à la publication de notifications d'événements Amazon S3 sur une rubrique Amazon Simple Notification Service).

Rubriques

- Octroi d'autorisations pour appeler une AWS Lambda fonction
- Octroi d'autorisations pour la publication de messages dans une rubrique SNS ou une file d'attente SQS

Octroi d'autorisations pour appeler une AWS Lambda fonction

Amazon S3 publie des messages d'événement sur AWS Lambda en invoquant une fonction Lambda et en fournissant le message d'événement comme argument.

Lorsque vous utilisez la console Amazon S3 pour configurer les notifications d'événements sur un compartiment Amazon S3 pour une fonction Lambda, la console définit les autorisations nécessaires au niveau de la fonction Lambda. Amazon S3 dispose ainsi d'autorisations pour appeler la fonction à partir du compartiment. Pour plus d'informations, consultez Activation et configuration des notifications d'événements à l'aide de la console Amazon S3.

Vous pouvez également accorder à Amazon S3 des autorisations AWS Lambda pour appeler votre fonction Lambda. Pour plus d'informations, consultez Tutoriel : Utilisation AWS Lambda avec Amazon S3 dans le Guide du AWS Lambda développeur.

Octroi d'autorisations pour la publication de messages dans une rubrique SNS ou une file d'attente SQS

Pour accorder à Amazon S3 l'autorisation de publier des messages sur la rubrique SNS ou la file d'attente SQS, associez une politique AWS Identity and Access Management (IAM) à la rubrique SNS ou à la file d'attente SQS de destination.

Pour afficher un exemple montrant comment associer une stratégie à une rubrique SNS ou à une file d'attente SQS, veuillez consulter Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS). Pour en savoir plus sur les autorisations, consultez les rubriques suivantes :

- Cas d'utilisation du contrôle des accès Amazon SNS dans le Manuel du développeur Amazon Simple Notification Service
- Identity and Access Management dans Amazon SQS dans le Manuel du développeur Amazon Simple Queue Service

Stratégie IAM pour une rubrique SNS de destination

Voici un exemple de stratégie AWS Identity and Access Management (IAM) que vous associez à la rubrique SNS de destination. Pour savoir comment utiliser cette politique pour configurer une rubrique Amazon SNS de destination pour les notifications d'événements, veuillez consulter Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS).

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

Stratégie IAM pour une file d'attente SQS de destination

Voici un exemple de stratégie IAM que vous associez à la file d'attente SQS de destination. Pour savoir comment utiliser cette politique pour configurer une file d'attente Amazon SQS de destination pour les notifications d'événements, veuillez consulter Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS).

Pour utiliser cette politique, vous devez mettre à jour l'ARN de la file d'attente Amazon SQS, le nom du bucket et l'ID du propriétaire du Compte AWS bucket.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "arn:aws:sqs:Region:account-id:queue-name",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

Qu'il s'agisse de la stratégie IAM Amazon SNS ou de la stratégie IAM Amazon SQS, vous pouvez spécifier la condition `StringLike` en lieu et place de la condition `ArnLike` dans la stratégie.

Lorsque `ArnLike` est utilisé, les parties partition, service, ID de compte, type de ressource et ID de ressource partiel de l'ARN doivent correspondre exactement à l'ARN dans le contexte de la demande. Seuls la région et le chemin de ressource permettent une correspondance partielle.

Lorsque `StringLike` est utilisé à la place de `ArnLike`, la correspondance ignore la structure de l'ARN et permet une correspondance partielle, quelle que soit la partie qui a été marquée par un caractère générique. Pour plus d'informations, consultez Éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:bucket-name" }
  }
```

AWS KMS politique clé

Si la file d'attente SQS ou les sujets SNS sont chiffrés à l'aide d'une clé gérée par le client AWS Key Management Service (AWS KMS), vous devez accorder au principal du service Amazon S3 l'autorisation d'utiliser les sujets ou la file d'attente chiffrés. Ajoutez l'instruction suivante à la politique de la clé gérée par le client afin d'accorder l'autorisation au principal de service Amazon S3.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Pour plus d'informations sur les politiques AWS KMS clés, consultez la section Utilisation des politiques clés AWS KMS dans le Guide du AWS Key Management Service développeur.

Pour plus d'informations sur l'utilisation du chiffrement côté serveur AWS KMS pour Amazon SQS et Amazon SNS, consultez ce qui suit :

- Gestion des clés dans le Manuel du développeur Amazon Simple Notification Service.
- Gestion des clés dans le Manuel du développeur Amazon Simple Queue Service.
- Chiffrement des messages publiés vers Amazon SNS avec AWS KMS dans le Blog sur le calcul AWS .

Activation et configuration des notifications d'événements à l'aide de la console Amazon S3

Vous pouvez autoriser le compartiment Amazon S3 à envoyer un message de notification à une destination chaque fois que ces événements se produisent. Cette section explique comment utiliser la console Amazon S3 pour activer les notifications d'événement. Pour plus d'informations sur l'utilisation des notifications d'événements avec les AWS SDK et les API REST d'Amazon S3, consultezConfiguration des notifications d'événements par programmation.

Prérequis : avant de pouvoir activer les notifications d'événements pour votre compartiment, vous devez configurer l'un des types de destination, puis configurer les autorisations. Pour plus d'informations, consultez Destinations d'événements prises en charge et Octroi d'autorisations pour la publication de messages de notification d'événement vers une destination.

> **Note**
>
> Les files d'attente Amazon Simple Queue Service FIFO (premier entré, premier sorti) ne sont pas prises en charge en tant que destination des notifications d'événements Amazon S3. Pour envoyer une notification concernant un événement Amazon S3 à une file d'attente FIFO Amazon SQS, vous pouvez utiliser Amazon. EventBridge Pour plus d'informations, consultez Activation d'Amazon EventBridge.

Rubriques

- Activation des notifications Amazon SNS, Amazon SQS ou Lambda à l'aide de la console Amazon S3

Activation des notifications Amazon SNS, Amazon SQS ou Lambda à l'aide de la console Amazon S3

Pour activer et configurer des notifications d'événements pour un compartiment S3

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer les événements.
3. Choisissez Propriétés.

4. Accédez à la section Event Notifications (Notifications d'événements) et choisissez Create event notification (Créer une notification d'événements).
5. Dans la section General configuration (Configuration générale) spécifiez le nom d'événement descriptif pour votre notification d'événements. Vous pouvez également spécifier un préfixe et un suffixe pour limiter les notifications aux objets dont les clés se terminent par les caractères spécifiés.

   a. Saisissez une description pour Event name (Nom de l'événement).

      Si vous ne saisissez pas de nom, un identificateur global unique (GUID) sera généré et utilisé pour le nom.

   b. (Facultatif) Pour filtrer les notifications d'événements par préfixe, saisissez un préfixe.

      Par exemple, vous pouvez configurer un filtre de préfixe de sorte à recevoir uniquement des notifications lorsque des fichiers sont ajoutés à un dossier spécifique (par exemple, `images/`).

   c. (Facultatif) Pour filtrer les notifications d'événements par suffixe, saisissez un suffixe.

      Pour plus d'informations, consultez Configuration des notifications d'événement à l'aide du filtrage des noms de clé d'objet.
6. Dans la section Event types (Types d'événements), sélectionnez un ou plusieurs types d'événements pour lesquels vous souhaitez recevoir des notifications.

   Pour obtenir la liste des différents types d'événements, veuillez consulter Types d'événements pris en charge pour SQS, SNS et Lambda.
7. Dans la section Destination choisissez la destination de notification d'événements.

   > **Note**
   >
   > Avant de publier des notifications d'événements, vous devez accorder au principal Amazon S3 les autorisations requises pour qu'il puisse appeler l'API pertinente. Il peut ainsi publier des notifications vers une fonction Lambda, une rubrique SNS ou une file d'attente SQS.

   a. Sélectionnez le type de destination : Lambda Function (Fonction Lambda), SNS Topic (Rubrique SNS) ou SQS Queue (File d'attente SQS).

b. Après avoir choisi votre type de destination, choisissez une fonction, une rubrique ou une file d'attente dans la liste.

c. Si vous préférez spécifier un Amazon Resource Name (ARN), utilisez l'option Saisir l'ARN.

Pour plus d'informations, consultez Destinations d'événements prises en charge.

8. Choisissez Enregistrer les modifications. Amazon S3 envoie alors un message de test à la destination de notification d'événements.

Configuration des notifications d'événements par programmation

Par défaut, les notifications sont désactivées pour tous les types d'événements. Par conséquent, la sous-ressource notification stocke initialement une configuration vide.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Pour activer les notifications pour certains types d'événements, vous devez remplacer le fichier XML par la configuration appropriée, qui identifie les types d'événements que doit publier Amazon S3, ainsi que leurs destinations. Vous devez ajouter une configuration XML pour chaque destination.

Publication des messages d'événement dans une file d'attente SQS

Pour définir une file d'attente SQS comme destination de notification pour un ou plusieurs types d'événements, ajoutez l'élément `QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
     ...
  </QueueConfiguration>
   ...
</NotificationConfiguration>
```

Publication des messages d'événement dans une rubrique SNS

Pour définir une rubrique SNS comme destination de notification pour des types d'événements spécifiques, ajoutez l'élément `TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>optional-id-string</Id>
     <Topic>sns-topic-arn</Topic>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </TopicConfiguration>
   ...
</NotificationConfiguration>
```

Pour appeler la AWS Lambda fonction et fournir un message d'événement en tant qu'argument

Pour définir une fonction Lambda comme destination de notification pour des types d'événements spécifiques, ajoutez l'élément `CloudFunctionConfiguration`.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>optional-id-string</Id>
     <CloudFunction>cloud-function-arn</CloudFunction>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </CloudFunctionConfiguration>
   ...
</NotificationConfiguration>
```

Suppression de toute les notifications configurées pour un compartiment

Pour supprimer toutes les notifications configurées pour un compartiment, enregistrez un élément `<NotificationConfiguration/>` vide dans la sous-ressource notification.

Lorsque Amazon S3 détecte un événement qui appartient au type spécifié, il publie un message contenant les informations relatives à cet événement. Pour plus d'informations, consultez Structure des messages d'événements.

Pour plus d'informations sur la configuration des notifications d'événements, consultez les rubriques suivantes :

- Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS).
- Configuration des notifications d'événement à l'aide du filtrage des noms de clé d'objet

Démonstration : configuration d'un compartiment pour les notifications (rubrique SNS ou file d'attente SQS)

Vous pouvez recevoir des notifications Amazon S3 à l'aide d'Amazon Simple Notification Service (Amazon SNS) ou d'Amazon Simple Queue Service (Amazon SQS). Dans le cadre de cette démonstration, vous allez ajouter une configuration de notification à votre compartiment à l'aide d'une rubrique Amazon SNS et d'une file d'attente Amazon SQS.

> **Note**
>
> Les files d'attente Amazon Simple Queue Service FIFO (premier entré, premier sorti) ne sont pas prises en charge en tant que destination des notifications d'événements Amazon S3. Pour envoyer une notification concernant un événement Amazon S3 à une file d'attente FIFO Amazon SQS, vous pouvez utiliser Amazon. EventBridge Pour plus d'informations, consultez Activation d'Amazon EventBridge.

Rubriques

- Résumé de la procédure détaillée
- Étape 1 : Créer une file d'attente Amazon SQS
- Étape 2 : Créer une rubrique Amazon SNS
- Étape 3 : Ajouter une configuration de notifications à votre compartiment
- Étape 4 : Tester la configuration

Résumé de la procédure détaillée

Cette démonstration vous explique comment :

- Publier des événements de type `s3:ObjectCreated:*` dans une file d'attente Amazon SQS.
- Publier des événements de type `s3:ReducedRedundancyLostObject` dans une rubrique Amazon SNS.

Pour en savoir plus sur la configuration des notifications, consultez Utilisation d'Amazon SQS, Amazon SNS et Lambda.

Vous pouvez suivre toutes ces étapes via la console sans écrire de code. En outre, des exemples de code utilisant AWS des SDK pour Java et .NET sont également fournis pour vous aider à ajouter des configurations de notification par programmation.

La procédure comprend les étapes suivantes :

1. Créez une file d'attente Amazon SQS.

   Via la console Amazon SQS, créez un file d'attente. Vous pouvez accéder par programmation à tous les messages qu'Amazon S3 envoie vers la file d'attente. Mais dans le cadre de cette procédure détaillée, vous vérifierez tous les messages de notification sur la console.

   Vous associez une stratégie d'accès à la file d'attente afin d'autoriser Amazon S3 à publier des messages.
2. Créez une rubrique Amazon SNS.

   À l'aide de la console Amazon SNS, créez une rubrique SNS et abonnez-y. Ainsi, tout événement publié dans cette rubrique vous est signalé. Vous définissez l'e-mail comme protocole de communication. Lorsque vous créez une rubrique, Amazon SNS vous envoie un e-mail. Vous suivez le lien contenu dans l'e-mail pour confirmer votre abonnement à la rubrique.

   Vous associez une stratégie d'accès à la rubrique, afin d'autoriser Amazon S3 à publier des messages.
3. Ajouter une configuration de notifications à un compartiment.

Étape 1 : Créer une file d'attente Amazon SQS

Suivez les étapes pour créer une file d'attente Amazon Simple Queue Service (Amazon SQS) et vous y abonner.

1. À l'aide de la console Amazon SQS, créez une file d'attente. Pour obtenir des instructions, veuillez consulter Démarrez avec Amazon SQS dans le Manuel du développeur Amazon Simple Queue Service.
2. Remplacez la stratégie d'accès qui est associée à la file d'attente par la stratégie ci-après.

a. Dans la console Amazon SQS, dans la liste Queues (Files d'attente), sélectionnez le nom de la file d'attente.

b. Dans l'onglet Access policy (Stratégie d'accès), choisissez Edit (Modifier).

c. Remplacez la stratégie d'accès qui est associée à la file d'attente. Indiquez votre ARN Amazon SQS, le nom du compartiment source et l'ID de compte du propriétaire du compartiment.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "SQS-queue-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

d. Choisissez Enregistrer.

3. (Facultatif) Si le chiffrement côté serveur est activé AWS Key Management Service avec AWS KMS() dans la file d'attente Amazon SQS ou dans la rubrique Amazon SNS, ajoutez la politique suivante à la clé de chiffrement symétrique gérée par le client associée.

   Vous devez ajouter la politique à une clé gérée par le client, car vous ne pouvez pas modifier la clé gérée par AWS pour Amazon SQS ou Amazon SNS.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Pour plus d'informations sur l'utilisation de SSE pour Amazon SQS et Amazon SNS AWS KMS avec, consultez ce qui suit :

- Gestion des clés dans le Manuel du développeur Amazon Simple Notification Service.
- Gestion des clés dans le Manuel du développeur Amazon Simple Queue Service.

4. Notez l'ARN de la file d'attente.

   La file d'attente SQS que vous avez créée est une autre ressource dans votre Compte AWS. Elle possède un Amazon Resource Name (ARN) unique. Vous avez besoin de cet ARN à la prochaine étape. L'ARN présente dans le format suivant :

   ```
   arn:aws:sqs:aws-region:account-id:queue-name
   ```

Étape 2 : Créer une rubrique Amazon SNS

Suivez ces étapes pour créer une rubrique Amazon SNS et vous y abonner.

1. Créez une rubrique à partir de la console Amazon SNS. Des instructions sont disponibles dans la section Création d'une rubrique Amazon SNS du Manuel du développeur Amazon Simple Notification Service.

2. Abonnez-vous à la rubrique. Dans le cadre de cet exercice, définissez l'e-mail comme protocole de communication. Pour obtenir des instructions, veuillez consulter Abonnement à une rubrique Amazon SNS dans le Manuel du développeur Amazon Simple Notification Service.

   Vous recevez un e-mail vous invitant à confirmer votre abonnement à la rubrique. Confirmez votre abonnement.

3. Remplacez la stratégie d'accès associée à la rubrique par la stratégie ci-après. Indiquez l'ARN de votre rubrique SNS, le nom du compartiment et l'ID de compte du propriétaire du compartiment.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:bucket-name"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

4. Notez l'ARN de la rubrique.

   La rubrique SNS que vous avez créée est une autre ressource de votre Compte AWS choix, et elle possède un ARN unique. Vous aurez besoin de cet ARN à la prochaine étape. L'ARN présentera le format suivant :

```
arn:aws:sns:aws-region:account-id:topic-name
```

Étape 3 : Ajouter une configuration de notifications à votre compartiment

Vous pouvez activer les notifications de compartiment à l'aide de la console Amazon S3 ou par programmation à l'aide AWS de kits de développement logiciel (SDK). Choisissez l'une des deux méthodes pour configurer les notifications de votre compartiment. Cette section fournit des exemples de code à utiliser avec les kits AWS SDK pour Java et .NET.

Option A : Activer des notifications d'un compartiment via la console

Dans la console Amazon S3, ajoutez une configuration de notifications demandant à Amazon S3 de :

- Publier les événements du type All objet create events (Tous les événements de création d'objet) dans votre file d'attente Amazon SQS.
- Publier les événements du type Object in RRS lost (Objet dans le stockage Reduced Redundancy Storage (RRS) perdu) dans votre rubrique Amazon SNS.

Une fois que vous avez enregistré la configuration de notifications, Amazon S3 publie un message de test, que vous recevez par e-mail.

Pour obtenir des instructions, veuillez consulter Activation et configuration des notifications d'événements à l'aide de la console Amazon S3.

Option B : activer les notifications sur un bucket à l'aide des AWS SDK

.NET

L'exemple de code C# ci-dessous fournit une liste complète de codes qui permet d'ajouter une configuration de notifications à un compartiment. Vous devez mettre à jour le code et fournir le nom de votre compartiment, ainsi que l'ARN de la rubrique SNS. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
```

```
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string snsTopic = "*** SNS topic ARN ***";
        private const string sqsQueue = "*** SQS topic ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            EnableNotificationAsync().Wait();
        }

        static async Task EnableNotificationAsync()
        {
            try
            {
               PutBucketNotificationRequest request = new
 PutBucketNotificationRequest
                {
                    BucketName = bucketName
                };

                TopicConfiguration c = new TopicConfiguration
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic
                };
                request.TopicConfigurations = new List<TopicConfiguration>();
                request.TopicConfigurations.Add(c);
                request.QueueConfigurations = new List<QueueConfiguration>();
                request.QueueConfigurations.Add(new QueueConfiguration()
                {
                    Events = new List<EventType> { EventType.ObjectCreatedPut },
                    Queue = sqsQueue
                });
```

```
                PutBucketNotificationResponse response = await
 client.PutBucketNotificationAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' ",
 e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown error encountered on server.
 Message:'{0}' ", e.Message);
            }
        }
    }
}
```

Java

L'exemple suivant montre comment ajouter une configuration de notification à un compartiment. Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.EnumSet;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        Regions clientRegion = Regions.DEFAULT_REGION;
        String snsTopicARN = "*** SNS Topic ARN ***";
        String sqsQueueARN = "*** SQS Queue ARN ***";
```

```
        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();
            BucketNotificationConfiguration notificationConfiguration = new
 BucketNotificationConfiguration();

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                    new TopicConfiguration(snsTopicARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
            notificationConfiguration.addConfiguration("sqsQueueConfig",
                    new QueueConfiguration(sqsQueueARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Create the notification configuration request and set the bucket
 notification
            // configuration.
            SetBucketNotificationConfigurationRequest request = new
 SetBucketNotificationConfigurationRequest(
                    bucketName, notificationConfiguration);
            s3Client.setBucketNotificationConfiguration(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Étape 4 : Tester la configuration

Vous pouvez désormais tester la configuration. Pour ce faire, chargez un objet vers votre compartiment et vérifiez la notification d'événement dans la console Amazon SQS. Pour obtenir

des instructions, veuillez consulter Recevez et supprimez votre message dans la section « Mise en route » du Manuel du développeur Amazon Simple Queue Service.

Configuration des notifications d'événement à l'aide du filtrage des noms de clé d'objet

Lorsque vous configurez une notification d'événements Amazon S3, vous devez spécifier quels types d'événements Amazon S3 pris en charge entraînent l'envoi de la notification par Amazon S3. Si un type d'événement que vous n'avez pas spécifié se produit dans votre compartiment S3, Amazon S3 n'envoie pas de notification.

Vous pouvez configurer les notifications pour qu'elles soient filtrées en fonction du préfixe et du suffixe du nom de clé des objets. Par exemple, vous pouvez définir une configuration afin de ne recevoir une notification uniquement lorsque des fichiers image comportant l'extension de nom de fichier « `.jpg` » sont ajoutés à un compartiment. Vous pouvez également avoir une configuration qui envoie une notification à une rubrique Amazon SNS lorsqu'un objet avec le préfixe « `images/` » est ajouté au compartiment, tandis que des notifications pour les objets ayant un préfixe « `logs/` » dans le même compartiment sont envoyées à une fonction. AWS Lambda

> **Note**
>
> Les filtres n'acceptent pas de caractère générique (« * ») comme préfixe ou suffixe. Si votre préfixe ou suffixe contient un espace, vous devez le remplacer par le caractère « + ». Si vous utilisez un autre caractère spécial dans la valeur du préfixe ou du suffixe, vous devez les entrer au format codé en URL (encodage-pourcent). Pour obtenir la liste complète des caractères spéciaux qui doivent être convertis en format codé URL lorsqu'ils sont utilisés dans un préfixe ou un suffixe pour les notifications d'événements, consultez Caractères adaptés.

Ces configurations de notifications en fonction du nom de clé d'objet peuvent être définies dans la console Amazon S3. Vous pouvez le faire en utilisant les API Amazon S3 via les AWS SDK ou directement les API REST. Pour en savoir plus sur l'utilisation de l'interface utilisateur de la console pour définir une configuration de notification pour un compartiment, veuillez consulter Activation et configuration des notifications d'événements à l'aide de la console Amazon S3.

Amazon S3 stocke la configuration des notifications au format XML dans la sous-ressource notification associée à un compartiment, comme décrit dans Utilisation d'Amazon SQS, Amazon SNS et Lambda. Pour définir les règles de filtrage des notifications en fonction du préfixe ou du suffixe

du nom clé de l'objet, utilisez la structure XML `Filter`. Pour plus d'informations sur la structure XML `Filter`, consultez PUT Bucket notification dans la Référence d'API Amazon Simple Storage Service.

Lorsque les configurations de notifications utilisent `Filter`, les règles de filtrage ne doivent pas contenir de préfixes qui se chevauchent, de suffixes qui se chevauchent, ni de préfixes et suffixes qui se chevauchent. Les sections suivantes contiennent des exemples de configurations de notification valides avec filtrage des noms de clé d'objet. Elles proposent également des exemples de configurations de notification non valides en raison du chevauchement des préfixes et des suffixes.

Rubriques

- Exemples corrects de configurations des notifications avec filtrage par nom de clé d'objet
- Exemples de configurations des notifications avec chevauchement incorrect des préfixes et des suffixes

Exemples corrects de configurations des notifications avec filtrage par nom de clé d'objet

La configuration des notifications ci-après comporte une configuration de file d'attente qui identifie une file d'attente Amazon SQS dans laquelle Amazon S3 doit publier les événements de type `s3:ObjectCreated:Put`. Les événements sont publiés chaque fois qu'un objet présentant un préfixe `images/` et un suffixe `jpg` est ajouté dans un compartiment via une demande PUT.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>images/</Value>
              </FilterRule>
              <FilterRule>
                  <Name>suffix</Name>
                  <Value>jpg</Value>
              </FilterRule>
          </S3Key>
      </Filter>
      <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
      <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
```

```
</NotificationConfiguration>
```

La configuration de notifications ci-dessous comporte plusieurs préfixes qui ne se chevauchent pas. La configuration indique que les notifications associées aux demandes PUT du dossier `images/` sont envoyées vers une file d'attente A (queue-A), alors que les notifications associées aux demandes PUT du dossier `logs/` sont envoyées vers une file d'attente B (queue-B).

```
<NotificationConfiguration>
  <QueueConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images/</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
     <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
  <QueueConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>logs/</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
     <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

La configuration de notifications ci-dessous comporte plusieurs suffixes qui ne se chevauchent pas. La configuration indique que toutes les images `.jpg` nouvellement ajoutées au compartiment sont traitées par la fonction Lambda cloud-function-A, et que toutes les images `.png` nouvellement ajoutées sont traitées par la fonction cloud-function-B. Les suffixes `.png` et `.jpg` ne se chevauchent pas, même si leur dernière lettre est identique. Si une chaîne donnée peut se terminer par deux

suffixes, on considère que les deux suffixes se chevauchent. Une chaîne ne pouvant se terminer par `.png` et `.jpg`, les suffixes de cet exemple de configuration ne se chevauchent pas.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

Lorsque vos configurations de notifications utilisent `Filter`, les règles de filtrage associées à des types d'événements identiques ne doivent pas contenir de préfixes qui se chevauchent. Ils ne peuvent le faire que si les préfixes qui se chevauchent sont utilisés avec des suffixes qui ne se chevauchent pas. Dans l'exemple de configuration ci-dessous, nous voyons que des objets créés avec un préfixe commun, mais dont les suffixes ne se chevauchent pas, peuvent être envoyés vers différentes destinations.

```
<NotificationConfiguration>
```

```
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

Exemples de configurations des notifications avec chevauchement incorrect des préfixes et des suffixes

Lorsque vos configurations de notifications utilisent `Filter`, dans la majorité des cas, les règles de filtrage associées à des types d'événements identiques ne doivent pas contenir de préfixes qui se

chevauchent, de suffixes qui se chevauchent, ni de combinaisons de préfixes et de suffixes qui se chevauchent. Elles peuvent toutefois comporter des préfixes qui se chevauchent, à condition que les suffixes ne se chevauchent pas. Pour obtenir un exemple, consultez Configuration des notifications d'événement à l'aide du filtrage des noms de clé d'objet.

En revanche, il est possible d'utiliser des filtres par nom de clé d'objet qui se chevauchent lorsque les types d'événements sont différents. Par exemple, vous pouvez créer une configuration de notifications qui utilise le préfixe `image/` pour le type d'événement `ObjectCreated:Put` et le préfixe `image/` pour le type d'événement `ObjectRemoved:*`.

Que vous utilisiez la console Amazon S3 ou l'API, si vous tentez d'enregistrer une configuration des notifications comportant des filtres de nom non valides qui se chevauchent pour les mêmes types d'événements, vous obtenez un message d'erreur. Cette section présente des exemples de configurations des notifications incorrectes, en raison du chevauchement des filtres de nom.

Toute règle de configuration des notifications existante est supposée comporter un préfixe et un suffixe par défaut qui correspondent respectivement à n'importe quel autre préfixe et suffixe. La configuration des notifications ci-dessous est incorrecte, car elle comporte des préfixes qui se chevauchent. Plus précisément, le préfixe racine chevauche n'importe quel autre préfixe. Cela est également vrai si, dans cet exemple, vous utilisiez un suffixe en lieu et place d'un préfixe. Le suffixe racine chevaucherait également n'importe quel autre suffixe.

```
<NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-two</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>prefix</Name>
                     <Value>images</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

La configuration des notifications ci-dessous est incorrecte, car elle comporte des suffixes qui se chevauchent. Si une chaîne donnée peut se terminer par deux suffixes, on considère que les deux suffixes se chevauchent. Une chaîne peut se terminer par `jpg` et `pg`. Ainsi, les suffixes se chevauchent. Il en va de même pour les préfixes. Si une chaîne donnée peut commencer par deux préfixes, on considère que les deux préfixes se chevauchent.

```
 <NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>pg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration
```

La configuration des notifications ci-dessous est incorrecte, car elle comporte des préfixes et des suffixes qui se chevauchent.

```
<NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
```

```
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>jpg</Value>
                </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
             </S3Key>
         </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

Structure des messages d'événements

Le message de notification qu'envoie Amazon S3 pour publier un événement est au format JSON.

Pour obtenir une présentation générale et des instructions sur la configuration des notifications d'événement, veuillez consulter Notifications d'événements Amazon S3.

Cet exemple illustre la version 2.2 de la structure JSON de notification d'événement. Amazon S3 utilise les versions 2.1, 2.2 et 2.3 de cette structure d'événement. Amazon S3 utilise la version 2.2 pour les notifications d'événements de réplication entre Régions. Il utilise la version 2.3 pour le cycle de vie S3, S3 Intelligent-Tiering, la liste ACL d'objet, le balisage d'objets et les événements de suppression de restauration d'objets. Ces versions contiennent des informations supplémentaires spécifiques à ces opérations. Les versions 2.2 et 2.3 sont par ailleurs compatibles avec la version 2.1, qu'Amazon S3 utilise actuellement pour tous les autres types de notifications d'événements.

```
{
```

```
  "Records":[
     {
        "eventVersion":"2.2",
        "eventSource":"aws:s3",
        "awsRegion":"us-west-2",
        "eventTime":"The time, in ISO-8601 format, for example,
1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request",
        "eventName":"event-type",
        "userIdentity":{
           "principalId":"Amazon-customer-ID-of-the-user-who-caused-the-event"
        },
        "requestParameters":{
           "sourceIPAddress":"ip-address-where-request-came-from"
        },
        "responseElements":{
           "x-amz-request-id":"Amazon S3 generated request ID",
           "x-amz-id-2":"Amazon S3 host that processed the request"
        },
        "s3":{
           "s3SchemaVersion":"1.0",
           "configurationId":"ID found in the bucket notification configuration",
           "bucket":{
              "name":"bucket-name",
              "ownerIdentity":{
                 "principalId":"Amazon-customer-ID-of-the-bucket-owner"
              },
              "arn":"bucket-ARN"
           },
           "object":{
              "key":"object-key",
              "size":"object-size in bytes",
              "eTag":"object eTag",
              "versionId":"object version if bucket is versioning-enabled, otherwise
null",
              "sequencer": "a string representation of a hexadecimal value used to
determine event sequence, only used with PUTs and DELETEs"
           }
        },
        "glacierEventData": {
           "restoreEventData": {
              "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for
example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
              "lifecycleRestoreStorageClass": "Source storage class for restore"
           }
```

```
        }
      }
   ]
}
```

Notez les éléments suivants à propos de la structure des messages d'événement :

- La valeur de clé `eventVersion` contient une version majeure et une version mineure au format `<major>.<minor>`.

  La version majeure est incrémentée si Amazon S3 apporte à la structure de l'événement une modification qui n'est pas rétrocompatible. Cela inclut la suppression d'un champ JSON déjà présent ou la modification de la représentation du contenu d'un champ (par exemple, un format de date).

  La version mineure est incrémentée si Amazon S3 ajoute de nouveaux champs à la structure de l'événement. Cela peut se produire si de nouvelles informations sont fournies pour certains ou tous les événements existants. Cela peut également se produire si de nouvelles informations sont fournies sur les types d'événements nouvellement introduits. Les applications doivent ignorer les nouveaux champs pour être compatibles avec de nouvelles versions mineures de la structure de l'événement.

  Si de nouveaux types d'événement sont introduits mais que la structure de l'événement n'est pas modifiée, la version de l'événement ne change pas.

  Pour vous assurer que vos applications puissent analyser correctement la structure de l'événement, nous vous recommandons d'effectuer une comparaison égal à sur le numéro de version majeure. Pour vous assurer que les champs attendus par votre application sont présents, nous vous recommandons également d'effectuer une comparaison « greater-than-or-equal -to » sur la version mineure.
- Le `eventName` fait référence à la liste des types de notifications d'événements, mais ne contient pas le préfixe `s3:`.
- La valeur `responseElements` clé est utile si vous souhaitez suivre une demande en effectuant un suivi avec AWS Support. Les deux éléments `x-amz-request-id` et `x-amz-id-2` aident Amazon S3 à suivre une demande individuelle. Ces valeurs sont identiques à celles renvoyées par Amazon S3 dans la réponse à la demande qui initie les événements. Ainsi, ils peuvent être utilisés pour faire correspondre l'événement à la demande.

- La clé `s3` fournit les informations relatives à l'objet et au compartiment concernés par l'événement. La valeur du nom de clé d'objet est codée en URL. Par exemple, « fleur rouge.jpg » devient « fleur +rouge.jpg » (Amazon S3 renvoie « `application/x-www-form-urlencoded` » comme type de contenu dans la réponse).
- La clé `sequencer` permet de déterminer la séquence des événements. Les notifications d'événements n'arrivent pas nécessairement dans le même ordre dans lequel les événement se sont produits. Toutefois, les notifications provenant d'événements qui créent des objets (PUTs) et supprimer des objets qui contiennent un `sequencer`. Il peut être utilisé pour déterminer l'ordre dans lequel se sont produits les événements pour une clé d'objet donnée.

  Si vous comparez les chaînes `sequencer` provenant de deux notifications d'événements associées à la même clé d'objet, la notification d'événement qui présente la plus grande valeur `sequencer` hexadécimale correspond à l'événement le plus récent. Si vous utilisez des notifications d'événement pour gérer une base de données ou un index distincts de vos objets Amazon S3, nous vous recommandons de comparer et de stocker les valeurs `sequencer` à mesure que vous traitez les notifications associées à chaque événement.

  Notez ce qui suit :

  - Vous ne pouvez pas utiliser `sequencer` pour déterminer l'ordre des événements associés à différentes clés d'objet.
  - La longueur des valeurs de séquenceur est variable. Pour comparer ces valeurs, complétez d'abord à droite la valeur la plus courte à l'aide de zéros, puis effectuer une comparaison lexicographique.
- Seule la clé `glacierEventData` est visible pour les événements `s3:ObjectRestore:Completed`.
- La clé `restoreEventData` contient des attributs qui sont liés à votre demande de restauration.
- La clé `replicationEventData` n'est visible que pour les événements de réplication.
- La clé `intelligentTieringEventData` n'est visible que pour les événements S3 Intelligent-Tiering.
- La clé `lifecycleEventData` n'est visible que pour les événements de transition du cycle de vie S3.

Exemples de messages

Voici des exemples de messages de notification d'événement Amazon S3.

Message de test Amazon S3

Une fois que vous avez configuré une notification d'événement pour un compartiment, Amazon S3 envoie le message de test suivant.

```
{
   "Service":"Amazon S3",
   "Event":"s3:TestEvent",
   "Time":"2014-10-13T15:57:02.089Z",
   "Bucket":"bucketname",
   "RequestId":"5582815E1AEA5ADF",
   "HostId":"8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

Exemple de message envoyé lorsqu'un objet est créé à l'aide d'une requête PUT

Le message suivant est un exemple de message envoyé par Amazon S3 pour publier un événement `s3:ObjectCreated:Put`.

```
{
   "Records":[
      {
         "eventVersion":"2.1",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"AIDAJDPLRKLG7UEXAMPLE"
         },
         "requestParameters":{
            "sourceIPAddress":"127.0.0.1"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"mybucket",
```

```
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::mybucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
               "versionId":"096fKKXTRTtl3on89fVO.nfljtsv6qko",
               "sequencer":"0055AED6DCD90281E5"
            }
         }
      }
   ]
}
```

Pour connaître la définition de chaque préfixe d'identification IAM (par exemple AIDA, AROA, AGPA), consultez la section Identifiants IAM du Guide de l'utilisateur IAM.

En utilisant EventBridge

Amazon S3 peut envoyer des événements à Amazon EventBridge chaque fois que certains événements se produisent dans votre compartiment. Contrairement à d'autres destinations, vous n'avez pas besoin de sélectionner les types d'événements que vous souhaitez proposer. Une fois EventBridge activé, tous les événements ci-dessous sont envoyés à EventBridge. Vous pouvez utiliser des EventBridge règles pour acheminer les événements vers des cibles supplémentaires. La liste suivante répertorie les événements auxquels Amazon S3 est envoyé EventBridge.

Type d'événement	Description
Objet créé	Un objet a été créé. Le champ Reason de la structure du message d'événement indique quelle API S3 a été utilisée pour créer l'objet : PutObject, POST Object CopyObject, ou CompleteMultipartUpload.
Objet supprimé (DeleteObject)	Un objet a été supprimé.

Type d’événement	Description
Objet supprimé (expiration du cycle de vie)	Lorsqu'un objet est supprimé à l'aide d'un appel d'API S3, le champ motif est défini sur DeleteObject. Lorsqu'un objet est supprimé par une règle d'expiration du cycle de vie S3, le champ motif est défini sur Expiration du cycle de vie. Pour plus d’informations, consultez Objets en cours d’expiration. Lorsqu'un objet non versionné est supprimé ou lorsqu'un objet versionné est supprimé définitivement, le champ de type de suppression est défini sur Supprimé définitivement. Lorsqu'un marqueur de suppression est créé pour un objet versionné, le champ de type de suppression est défini sur Supprimer le marqueur créé. Pour plus d’informations, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.
Restauration d'un objet démarrée	Une restauration d'objets a été initiée à partir de la classe de stockage S3 Glacier ou S3 Glacier Deep Archive ou depuis le niveau S3 Intelligent-Tiering Archive Access ou Deep Archive Access. Pour plus d’informations, consultez Utilisation des objets archivés.
Restauration d'un objet terminée	Une restauration d'objets a été terminée.
Restauration d'un objet expiré	La copie temporaire d'un objet restauré à partir de S3 Glacier ou S3 Glacier Deep Archive a expiré et a été supprimée.
Classe de stockage d'objets modifiée	Un objet a été transféré vers une classe de stockage différente. Pour plus d’informations, consultez Transition des objets à l'aide du cycle de vie Amazon S3.
Niveau d'accès aux objets modifié	Un objet a été transféré vers le niveau S3 Intelligent-Tiering Archive Access ou Deep Archive Access. Pour plus d’informations, consultez Amazon S3 Intelligent Tiering.

Type d’événement	Description
Liste ACL d'un objet mise à jour	La liste de contrôle d'accès (ACL) d'un objet a été définie à l'aide de l' PutObjectACL. Un événement n'est pas généré lorsqu'une demande n'entraîne aucune modification de la liste ACL d'un objet. Pour plus d’informations, consultez Présentation de la liste de contrôle d'accès (ACL).
Balises d'objets ajoutées	Un ensemble de balises a été ajouté à un objet à l'aide de PutObjectTagging. Pour plus d’informations, consultez Catégorisation de votre stockage à l'aide de balises.
Balises d'objets supprimées	Toutes les balises ont été supprimées d'un objet à l'aide de DeleteObjectTagging. Pour plus d’informations, consultez Catégorisation de votre stockage à l'aide de balises.

> **Note**
>
> Pour plus d'informations sur la façon dont les types d'événements Amazon S3 sont mappés aux types d' EventBridge événements, consultez EventBridge Cartographie et résolution des problèmes sur Amazon.

Vous pouvez utiliser les notifications d'événements Amazon S3 EventBridge pour écrire des règles qui prennent des mesures lorsqu'un événement se produit dans votre compartiment. Par exemple, vous pouvez faire en sorte qu'il vous envoie une notification. Pour plus d'informations, consultez le contenu EventBridge du guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations sur les actions et les types de données avec lesquels vous pouvez interagir à l'aide de l' EventBridge API, consultez la référence d' EventBridge API Amazon dans la référence EventBridge d'API Amazon.

Pour plus d'informations sur les tarifs, consultez EventBridge les tarifs Amazon.

Rubriques

- EventBridge Autorisations Amazon
- Activation d'Amazon EventBridge

- EventBridge structure du message d'événement
- EventBridge Cartographie et résolution des problèmes sur Amazon

EventBridge Autorisations Amazon

Amazon S3 n'a pas besoin d'autorisations supplémentaires pour transmettre des événements à Amazon EventBridge.

Activation d'Amazon EventBridge

Vous pouvez activer Amazon EventBridge à l'aide de la console S3 AWS Command Line Interface (AWS CLI) ou de l'API REST Amazon S3.

Utilisation de la console S3

Pour activer la diffusion d' EventBridge événements dans la console S3.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer les événements.
3. Choisissez Propriétés.
4. Accédez à la section Notifications d'événements et recherchez la EventBridge sous-section Amazon. Choisissez Modifier.
5. Sous Envoyer des notifications à Amazon EventBridge pour tous les événements de ce compartiment, sélectionnez Activé.

> **Note**
>
> Après l'activation EventBridge, il faut environ cinq minutes pour que les modifications prennent effet.

À l'aide du AWS CLI

L'exemple suivant crée une configuration de notification de compartiment pour un compartiment sur `DOC-EXAMPLE-BUCKET1` lequel Amazon EventBridge est activé.

```
aws s3api put-bucket-notification-configuration --bucket example-s3-bucket1 --
notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Utilisation de l'API REST

Vous pouvez activer Amazon par programmation EventBridge sur un bucket en appelant l'API REST Amazon S3. Pour plus d'informations, consultez le PutBucketNotificationConfigurationmanuel Amazon Simple Storage Service API Reference.

L'exemple suivant montre le code XML utilisé pour créer une configuration de notification de compartiment avec Amazon EventBridge activé.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

Création de EventBridge règles

Une fois activé, vous pouvez créer des EventBridge règles Amazon pour certaines tâches. Par exemple, vous pouvez envoyer des notifications par e-mail lorsqu'un objet est créé. Pour un didacticiel complet, consultez Tutoriel : Envoyer une notification lors de la création d'un objet Amazon S3 dans le guide de EventBridge l'utilisateur Amazon.

EventBridge structure du message d'événement

Le message de notification qu'envoie Amazon S3 pour publier un événement est au format JSON. Lorsqu'Amazon S3 envoie un événement à Amazon EventBridge, les champs suivants sont présents.

- version – Actuellement 0 (zéro) pour tous les événements.
- id – Un UUID version 4 généré pour chaque événement.
- detail-type – Type d'événement qui est envoyé. Consultez En utilisant EventBridge pour obtenir la liste des types d'événements.
- source – Identifie le service à l'origine de l'événement.
- account (compte) – ID de Compte AWS à 12 chiffres du propriétaire du compartiment.
- time (heure) – Heure à laquelle l'événement s'est produit.
- region (Région) – Identifie la Région AWS du compartiment.

- resources (ressources) – Tableau JSON contenant l'Amazon Resource Name (ARN) du compartiment.
- detail (détail) – Un objet JSON qui contient les informations sur l'événement. Pour plus d'informations sur ce qui peut être inclus dans ce champ, consultez Champ de détail des messages d'événement.

Exemples de structure de messages d'événements

Vous trouverez ci-dessous des exemples de messages de notification d'événements Amazon S3 qui peuvent être envoyés à Amazon EventBridge.

Objet créé

```
{
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7ededc63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
      "sequencer": "617f08299329d189"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "PutObject"
  }
}
```

Objet supprimé (en utilisant DeleteObject)

```
{
  "version": "0",
  "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "1QW9g1Z99LUNbvaaYVpW9xDlOLU.qxgF",
      "sequencer": "617f0837b476e463"
    },
    "request-id": "0BH729840619AG5K",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "DeleteObject",
    "deletion-type": "Delete Marker Created"
  }
}
```

Objet supprimé (à l'aide de l'expiration du cycle de vie)

```
{
  "version": "0",
  "id": "ad1de317-e409-eba2-9552-30113f8d88e3",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
```

```
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
      "sequencer": "617b398000000000"
    },
    "request-id": "20EB74C14654DC47",
    "requester": "s3.amazonaws.com",
    "reason": "Lifecycle Expiration",
    "deletion-type": "Delete Marker Created"
  }
}
```

Restauration d'un objet terminée

```
{
  "version": "0",
  "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
  "detail-type": "Object Restore Completed",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "DOC-EXAMPLE-BUCKET1"
    },
    "object": {
```

```
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
    },
    "request-id": "189F19CB7FB1B6A4",
    "requester": "s3.amazonaws.com",
    "restore-expiry-time": "2021-11-13T00:00:00Z",
    "source-storage-class": "GLACIER"
  }
}
```

Champ de détail des messages d'événement

Le champ de détails contient un objet JSON avec des informations sur l'événement. Les champs suivants peuvent être présents dans le champ de détail.

- version – Actuellement 0 (zéro) pour tous les événements.
- bucket (compartiment) – Informations relatives au compartiment Amazon S3 impliqué dans l'événement.
- object (objet) – Informations relatives à l'objet Amazon S3 impliqué dans l'événement.
- request-id – ID de demande dans la réponse S3.
- demandeur — Compte AWS ID ou principal AWS de service du demandeur.
- source-ip-address— Adresse IP source de la requête S3. Présent uniquement pour les événements déclenchés par une demande S3.
- reason — Pour les événements Object Created, l'API S3 utilisée pour créer l'objet : PutObject, POST Object CopyObject, ou CompleteMultipartUpload. Pour les événements Object Deleted, ce paramètre est défini sur DeleteObjectlorsqu'un objet est supprimé par un appel d'API S3, ou sur Expiration du cycle de vie lorsqu'un objet est supprimé par une règle d'expiration du cycle de vie S3. Pour plus d'informations, consultez Objets en cours d'expiration.
- deletion-type – Pour les événements Objet supprimé, lorsqu'un objet non versionné est supprimé ou lorsqu'un objet versionné est supprimé définitivement, ce champ est défini sur Supprimé définitivement. Lorsqu'un marqueur de suppression est créé pour un objet versionné, ce champ est défini sur Supprimer le marqueur créé. Pour plus d'informations, consultez Suppression des versions d'objet d'un compartiment activé pour la gestion des versions.

- restore-expiry-time— Pour les événements Object Restore Completed, heure à laquelle la copie temporaire de l'objet sera supprimée de S3. Pour plus d'informations, consultez Utilisation des objets archivés.
- source-storage-class— Pour les événements Object Restore Initiated et Object Restore Completed, classe de stockage de l'objet en cours de restauration. Pour plus d'informations, consultez Utilisation des objets archivés.
- destination-storage-class— Pour les événements Object Storage Class Changed, nouvelle classe de stockage de l'objet. Pour plus d'informations, consultez Transition des objets à l'aide du cycle de vie Amazon S3.
- destination-access-tier— Pour les événements Object Access Tier Changed, nouveau niveau d'accès de l'objet. Pour plus d'informations, consultez Amazon S3 Intelligent Tiering.

EventBridge Cartographie et résolution des problèmes sur Amazon

Le tableau suivant décrit comment les types d'événements Amazon S3 sont mappés aux types d' EventBridge événements Amazon.

Type d'événement S3	Type de EventBridge détail Amazon
ObjectCreated:Mettre ObjectCreated:Publier ObjectCreated:Copier ObjectCreated:CompleteMultipartUpload	Objet créé
ObjectRemoved:Supprimer ObjectRemoved:DeleteMarkerCreated LifecycleExpiration:Supprimer LifecycleExpiration:DeleteMarkerCreated	Objet supprimé

Type d'événement S3	Type de EventBridge détail Amazon
ObjectRestore:Publier	Restauration d'un objet démarrée
ObjectRestore:Terminé	Restauration d'un objet terminée
ObjectRestore:Supprimer	Restauration d'un objet expiré
LifecycleTransition	Classe de stockage d'objets modifiée
IntelligentTiering	Niveau d'accès aux objets modifié
ObjectTagging:Mettre	Balises d'objets ajoutées
ObjectTagging:Supprimer	Balises d'objets supprimées
ObjectAcl:Mettre	Liste ACL d'un objet mise à jour

EventBridge Résolution des problèmes liés à Amazon

Pour plus d'informations sur le dépannage EventBridge, consultez la section Résolution des problèmes liés EventBridge à Amazon dans le guide de EventBridge l'utilisateur Amazon.

Utilisation de l’analytique et des informations

Vous pouvez utiliser l’analytique et les informations dans Amazon S3 pour comprendre, analyser et optimiser votre utilisation du stockage. Pour plus d'informations, consultez les rubriques ci-dessous.

Rubriques

Analyses Amazon S3 - Analyse de classe de stockage

Grâce à l'analyse de classe de stockage Analyses Amazon S3, vous pouvez analyser des modèles d'accès au stockage pour décider à quel moment transférer les données appropriées vers la classe de stockage qui convient. Cette nouvelle fonction Analyses Amazon S3 observe les modèles d'accès aux données pour vous aider à déterminer le moment où passer du stockage STANDARD moins fréquemment utilisé à la classe de stockage STANDARD_IA (« IA » correspondant à « infrequent access », soit accès peu fréquents). Pour plus d'informations sur les classes de stockage, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Une fois que l'analyse de classe de stockage a observé les modèles d'accès peu fréquents d'un ensemble filtré de données sur une période donnée, vous pouvez utiliser les résultats d'analyse pour vous aider à améliorer vos configurations de cycle de vie. Vous pouvez configurer l'analyse de classe de stockage pour analyser tous les objets d'un compartiment. Ou, vous pouvez configurer des filtres pour regrouper des objets pour l'analyse par préfixe commun (c'est-à-dire des objets qui ont des noms qui commencent par une chaîne courante), par balises d'objets, ou par préfixe et balises. Vous trouverez très probablement qu'un filtrage par groupes d'objets est le meilleur moyen de tirer parti de l'analyse de classe de stockage.

> **Important**
>
> L'analyse de classe de stockage fournit uniquement des recommandations pour les classes standard à standard – Accès peu fréquent.

Vous pouvez avoir plusieurs filtres d'analyse de classe de stockage par compartiment, jusqu'à 1 000, et vous recevrez une analyse distincte pour chaque filtre. Plusieurs configurations de filtres vous permettent d'analyser des groupes spécifiques d'objets pour améliorer vos configurations de cycle de vie qui transfèrent des objets vers STANDARD_IA.

L'analyse de classe de stockage affiche des visualisations de l'utilisation du stockage dans la console Amazon S3 qui sont mises à jour quotidiennement. Vous pouvez également exporter ces données d'utilisation quotidienne vers un compartiment S3 et les consulter dans un tableur ou à l'aide d'outils de business intelligence tels qu'Amazon QuickSight.

Il y a des coûts associés à l'analyse de la classe de stockage. Pour de plus amples informations sur la tarification, veuillez consulter Gestion et réplication Tarification Amazon S3.

Rubriques

Comment configurer une analyse de classe de stockage ?

Vous configurez l'analyse de classe de stockage en configurant les données d'objet que vous voulez analyser. Vous pouvez configurer l'analyse de classe de stockage pour effectuer les opérations suivantes :

- Analyser l'ensemble des contenus d'un compartiment.

  Vous recevrez une analyse pour tous les objets du compartiment.
- Analyser les objets regroupés par préfixe et balises.

  Vous pouvez configurer des filtres qui regroupent des objets pour l'analyse par préfixe, par balises d'objets ou par une combinaison de préfixe et de balises. Vous recevez une analyse distincte pour chaque filtre que vous configurez. Vous pouvez avoir plusieurs configurations de filtres par compartiment, jusqu'à 1 000.
- Exporter les données d'analyse.

  Lorsque vous configurez l'analyse de classe de stockage pour un compartiment ou un filtre, vous pouvez choisir que les données d'analyse exportées soient exportées dans un fichier chaque

jour. L'analyse du jour est ajoutée au fichier pour former un journal d'analyse historique du filtre configuré. Le fichier est mis à jour quotidiennement au lieu de destination de votre choix. Lors de la sélection des données à exporter, vous spécifiez un compartiment de destination et un préfixe de destination facultatif où le fichier est écrit.

Vous pouvez utiliser la console Amazon S3, l'API REST AWS CLI ou les AWS SDK pour configurer l'analyse des classes de stockage.

- Pour en savoir plus sur la configuration de l'analyse de classe de stockage dans la console Amazon S3, consultez Configuration d'une analyse de classe de stockage.
- Pour utiliser l'API Amazon S3, utilisez l'PutBucketAnalyticsConfigurationAPI REST, ou l'équivalent, à partir du AWS CLI ou AWS des SDK.

Comment utiliser l'analyse de classe de stockage ?

Vous utilisez l'analyse de classe de stockage pour observer vos modèles d'accès aux données dans le temps en vue de collecter des informations vous permettant d'améliorer la gestion du cycle de vie de votre stockage STANDARD_IA. Une fois qu'un filtre est configuré, vous commencez à voir l'analyse des données basée sur le filtre dans la console Amazon S3 dans les 24 à 48 heures. Toutefois, l'analyse de classe de stockage observe les modèles d'accès d'un ensemble filtré de données pendant 30 jours ou plus afin d'obtenir des informations pour l'analyse avant de fournir un résultat. L'analyse se poursuit après le résultat initial et met à jour le résultat à mesure que les modèles d'accès changent

Lorsque vous configurez un filtre pour la première fois, la console Simple Storage Service (Amazon S3) peut prendre du temps pour analyser vos données.

L'analyse de classe de stockage observe les modèles d'accès d'un ensemble filtré de données d'objet pendant 30 jours ou plus pour collecter suffisamment d'informations pour l'analyse. Une fois que l'analyse de classe de stockage a rassemblé suffisamment d'informations, un message indiquant que l'analyse a été complétée s'affiche dans la console Amazon S3.

Lors de l'analyse des objets faisant l'objet d'accès peu fréquents, la classe de stockage examine le jeu filtré d'objets regroupés en fonction de l'âge comme ils ont été téléchargés vers Amazon S3. L'analyse de classe de stockage détermine si la tranche d'âge fait l'objet d'accès peu fréquents en regardant les facteurs suivants de l'ensemble filtré de données :

- Objets de la classe de stockage STANDARD de plus de 128 Ko.

- Stockage total moyen à disposition par tranche d'âge.
- Nombre moyen d'octets transférés (et non la fréquence) par tranche d'âge.
- Les données d'exportation d'analyse incluent uniquement les demandes avec des données pertinentes pour l'analyse de classe de stockage. Cela peut entraîner des différences dans le nombre de demandes, et dans le total des octets de chargement et de demande par rapport à ce qui est affiché dans les métriques de stockage ou suivi dans vos propres système internes.
- Les demandes GET et PUT en échec ne sont pas comptabilisées pour l'analyse. Cependant, vous voyez les demandes en échec dans les métriques de stockage.

Quelle quantité de données a été extraite du stockage ?

La console Amazon S3 affiche la quantité de stockage de l’ensemble de données filtré qui a été récupérée pour la période d’observation.

Quel pourcentage de données a été extrait du stockage ?

La console Amazon S3 affiche également le pourcentage de stockage de l’ensemble de données filtré qui a été récupéré pour la période d’observation.

Comme indiqué précédemment dans cette rubrique, lorsque vous exécutez l'analyse pour les objets faisant l'objet d'accès peu fréquents, l'analyse de classe de stockage examine le jeu filtré d'objets regroupés en fonction de l'âge comme ils ont été chargés dans Amazon S3. L'analyse de classe de stockage utilise les tranches d'âge d'objets prédéfinies suivantes :

- Objets Amazon S3 ayant moins de 15 jours
- Objets Amazon S3 ayant entre 15 et 29 jours
- Objets Amazon S3 ayant entre 30 et 44 jours
- Objets Amazon S3 ayant entre 45 et 59 jours
- Objets Amazon S3 ayant entre 60 et 74 jours
- Objets Amazon S3 ayant entre 75 et 89 jours
- Objets Amazon S3 ayant entre 90 et 119 jours
- Objets Amazon S3 ayant entre 120 et 149 jours
- Objets Amazon S3 ayant entre 150 et 179 jours
- Objets Amazon S3 ayant entre 180 et 364 jours
- Objets Amazon S3 ayant entre 365 et 729 jours
- Objets Amazon S3 ayant 730 jours et plus

Généralement, il faut environ 30 jours d'observation des modes d'accès pour collecter suffisamment d'informations pour un résultat d'analyse. Cette opération peut durer plus de 30 jours, selon le modèle d'accès unique de vos données. Toutefois, une fois qu'un filtre est configuré, vous commencez à voir l'analyse des données basée sur le filtre dans la console Amazon S3 dans les 24 à 48 heures. Vous pouvez voir quotidiennement l'analyse de l'accès aux objets réparti par tranche d'âge d'objets dans la console Amazon S3.

Quelle quantité de données du stockage fait l'objet d'accès peu fréquents ?

La console Amazon S3 affiche les modèles d'accès regroupés par groupes d'âge d'objets prédéfinis. Le texte Frequently accessed (Accès fréquent) ou Infrequently accessed (Accès peu fréquent) est conçu comme une aide visuelle pour vous aider dans le processus de création du cycle de vie.

Comment exporter des données d'analyse de classe de stockage ?

Vous pouvez choisir que le rapport d'analyse de classe de stockage soit au format de fichier plat CSV (valeurs séparées par une virgule). Les rapports sont mis à jour quotidiennement et sont basés sur les filtres de tranche d'âge d'objet que vous configurez. Lorsque vous utilisez la console Amazon S3, vous pouvez choisir l'option d'exportation des rapports quand vous créez un filtre. Lors de la sélection des données à exporter, vous spécifiez un compartiment de destination et un préfixe de destination facultatif où le fichier est écrit. Vous pouvez exporter les données vers un compartiment de destination d'un autre compte. Le compartiment de destination doit se trouver dans la même Région que le compartiment que vous configurez pour être analysé.

Vous devez créer une politique de compartiment sur le compartiment de destination afin d'accorder des autorisations à Amazon S3 afin de vérifier à qui Compte AWS appartient le compartiment et d'écrire des objets dans le compartiment à l'emplacement défini. Pour un exemple de stratégie, consultez Accorder des autorisations pour l'inventaire S3 et les analyses S3..

Une fois les rapports d'analyse de classe de stockage configurés, vous commencez à obtenir le rapport exporté quotidiennement au bout de 24 heures. Passé ce délai, Amazon S3 continue de surveiller et de fournir des exportations quotidiennes.

Vous pouvez ouvrir le fichier CSV dans un tableur ou l'importer dans d'autres applications comme Amazon QuickSight. Pour plus d'informations sur l'utilisation des fichiers Amazon S3 avec Amazon QuickSight, consultez la section Création d'un ensemble de données à l'aide de fichiers Amazon S3 dans le guide de QuickSight l'utilisateur Amazon.

Les données du fichier exporté sont triées par date dans la tranche d'âge d'objets, comme illustré dans les exemples suivants. Si la classe de stockage est STANDARD, la ligne

contient également les données des colonnes `ObjectAgeForSIATransition` et `RecommendedObjectAgeForSIATransition`.

Date	ConfigId	Filter	StorageClass	ObjectAge	ObjectCount	DataUploaded_MB	Storage_MB	DataRetrieved_MB	GetRequestCount	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		

À la fin du rapport, la tranche d'âge d'objets est nommée ALL. Les lignes ALL contiennent les totaux cumulés de toutes les tranche d'âge pour cette journée, y compris pour les objets inférieurs à 128 Ko.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02426125	015-029
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03545875	015-029
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.0209529	015-029
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02304819	015-029
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03073092	015-029
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014

La section suivante décrit les colonnes utilisées dans le rapport.

Disposition du fichier exporté

Le tableau suivant décrit la disposition du fichier exporté.

Configuration d’une analyse de classe de stockage

Grâce à l'outil d'analyse de classe de stockage Amazon S3, vous pouvez analyser des modèles d'accès au stockage pour décider à quel moment transférer les données appropriées vers la classe de stockage qui convient. L'analyse de classe de stockage observe les modèles d'accès aux données pour vous aider à déterminer quand il faut passer du mode STANDARD au mode STANDARD_IA (« IA » correspondant à « infrequent access », soit accès peu fréquent). Pour plus d’informations sur STANDARD_IA, consultez la FAQ Amazon S3 et Utilisation des classes de stockage Simple Storage Service (Amazon S3).

Vous configurez l'analyse de classe de stockage en configurant les données d'objet que vous voulez analyser. Vous pouvez configurer l'analyse de classe de stockage pour effectuer les opérations suivantes :

- Analyser l'ensemble des contenus d'un compartiment.

  Vous recevrez une analyse pour tous les objets du compartiment.
- Analyser les objets regroupés par préfixe et balises.

Vous pouvez configurer des filtres qui regroupent des objets pour l'analyse par préfixe, par balises d'objets ou par une combinaison de préfixe et de balises. Vous recevez une analyse distincte pour chaque filtre que vous configurez. Vous pouvez avoir plusieurs configurations de filtres par compartiment, jusqu'à 1 000.

- Exporter les données d'analyse.

  Lorsque vous configurez l'analyse de classe de stockage pour un compartiment ou un filtre, vous pouvez choisir que les données d'analyse exportées soient exportées dans un fichier chaque jour. L'analyse du jour est ajoutée au fichier pour former un journal d'analyse historique du filtre configuré. Le fichier est mis à jour quotidiennement au lieu de destination de votre choix. Lors de la sélection des données à exporter, vous spécifiez un compartiment de destination et un préfixe de destination facultatif où le fichier est écrit.

Vous pouvez utiliser la console Amazon S3, l'API REST AWS CLI ou les AWS SDK pour configurer l'analyse des classes de stockage.

> **Important**
>
> L'analyse de classe de stockage ne fournit pas de recommandations pour les transitions vers les classes de stockage Unizone – Accès peu fréquent ou S3 Glacier Flexible Retrieval.
> Si vous souhaitez configurer l'analyse des classes de stockage pour exporter vos résultats sous forme de fichier .csv et que le compartiment de destination utilise le chiffrement de compartiment par défaut avec un AWS KMS key, vous devez mettre à jour la politique AWS KMS clé afin d'accorder à Amazon S3 l'autorisation de chiffrer le fichier .csv. Pour obtenir des instructions, consultez Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement.

Pour plus d’informations sur l’analytique, consultez Analyses Amazon S3 - Analyse de classe de stockage.

Utiliser la console S3.

Pour configurer une analyse de classe de stockage

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez configurer une analyse de classe de stockage.
3. Sélectionnez l'onglet Métriques.
4. Sous Storage Class Analysis (Analyse des classes de stockage), choisissez Create analytics configuration (Créer une configuration d'analyse).
5. Attribuez un nom au filtre. Pour analyser le compartiment entier, n'indiquez rien dans le champ Prefix (Préfixe).
6. Dans le champ Prefix (Préfixe), saisissez du texte pour le préfixe des objets que vous souhaitez analyser.
7. Pour ajouter une balise, choisissez Add tag (Ajouter une balise). Saisissez une clé et une valeur pour la balise. Vous pouvez entrer un préfixe et plusieurs balises.
8. Vous pouvez également choisir Enable (Activer) sous Export CSV (Exporter vers CSV) pour exporter des rapports d'analyse au format de fichier plat de valeurs séparées par une virgule (.csv). Choisissez un compartiment de destination dans lequel le fichier peut être stocké. Vous pouvez saisir un préfixe pour le compartiment de destination. Le compartiment de destination doit se trouver dans le même compartiment Région AWS que celui pour lequel vous configurez l'analyse. Le compartiment de destination peut se trouver dans un autre Compte AWS.

   Si le compartiment de destination du fichier .csv utilise le chiffrement de compartiment par défaut avec une clé KMS, vous devez mettre à jour la politique relative aux AWS KMS clés pour autoriser Amazon S3 à chiffrer le fichier .csv. Pour obtenir des instructions, consultez Octroi à Amazon S3 d'utiliser votre clé gérée par le client pour le chiffrement.
9. Choisissez Create configuration (Créer une configuration).

Amazon S3 crée une stratégie de compartiment sur le compartiment de destination qui accorde à Amazon S3 une autorisation en écriture. Cela lui permettra d'écrire les données d'exportation dans le compartiment.

Si une erreur se produit lorsque vous tentez de créer la stratégie de compartiment, des instructions s'affichent pour vous indiquer comment la résoudre. Par exemple, si vous avez choisi un compartiment de destination dans un autre Compte AWS et ne disposez pas des autorisations nécessaires pour lire et écrire dans la stratégie de compartiment, le message suivant s'affiche. Vous devez demander au propriétaire du compartiment de destination d'ajouter la stratégie de compartiment affichée au compartiment de destination. Si la stratégie n'est pas ajoutée au compartiment de destination, vous ne recevez pas les données d'exportation, car Amazon S3 n'est pas autorisé à écrire dans le compartiment de destination. Si le compartiment source est détenu par

un compte autre que celui de l'utilisateur actuel, l'ID de compte correct du compartiment source doit être remplacé dans la stratégie.

Pour plus d'informations sur les données exportées et le fonctionnement du filtre, consultez Analyses Amazon S3 - Analyse de classe de stockage.

Utilisation de l'API REST

Pour configurer l'analyse des classes de stockage à l'aide de l'API REST, utilisez le PutBucketAnalyticsConfiguration. Vous pouvez également utiliser l'opération équivalente avec les AWS SDK AWS CLI or.

Vous pouvez utiliser les API REST suivantes pour travailler avec l'analyse de classe de stockage :

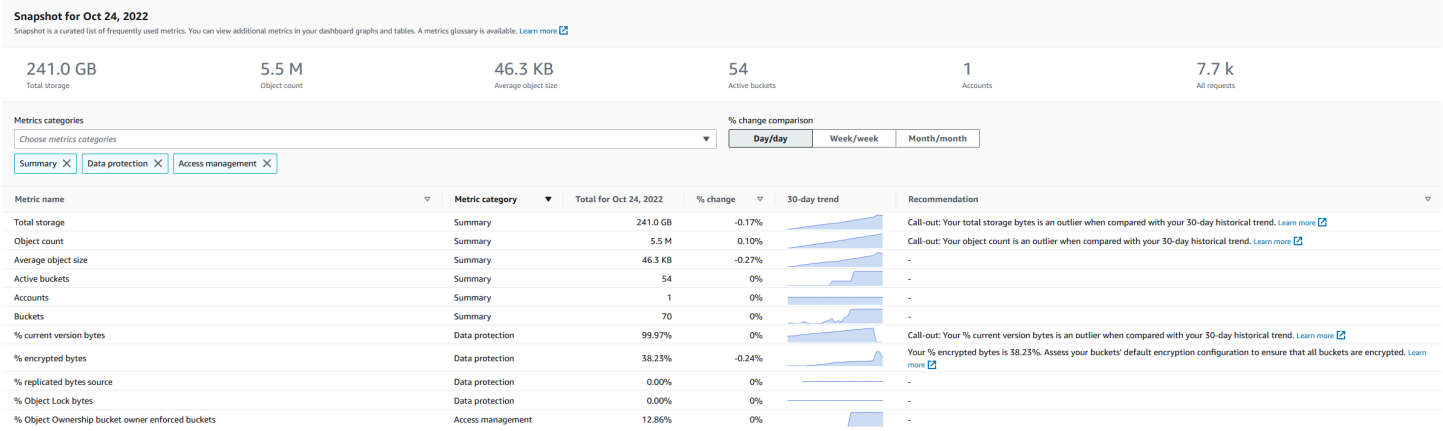
- DELETE Bucket Analytics configuration
- GET Bucket Analytics configuration
- List Bucket Analytics Configuration

Évaluer l'activité et l'utilisation de votre stockage avec Amazon S3 Storage Lens

Amazon S3 Storage Lens est une fonctionnalité d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur le stockage et l'activité des objets. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

Vous pouvez utiliser les métriques S3 Storage Lens pour générer des informations récapitulatives. Par exemple, vous pouvez connaître la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes dont la croissance est la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier des opportunités d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection des données et de gestion des accès, et améliorer les performances des charges de travail d'application. Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour abandonner les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3.

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.



Métriques et fonctionnalités S3 Storage Lens

S3 Storage Lens fournit un tableau de bord par défaut interactif qui est mis à jour quotidiennement. S3 Storage Lens préconfigure ce tableau de bord pour qu'il affiche le résumé des informations et des tendances pour l'ensemble de votre compte, et les met à jour quotidiennement dans la console S3. Les métriques de ce tableau de bord sont également résumées dans l'instantané du compte sur la page Buckets (compartiments). Pour de plus amples informations, veuillez consulter Tableau de bord par défaut.

Pour créer d'autres tableaux de bord et les délimiter par Régions AWS, compartiments S3 ou comptes (pour AWS Organizations), vous créez une configuration de tableau de bord S3 Storage Lens. Vous pouvez créer et gérer des configurations de tableau de bord S3 Storage Lens à l'aide de la console Amazon S3, d'AWS Command Line Interface (AWS CLI), des kits SDK AWS ou de l'API REST Amazon S3. Lorsque vous créez ou modifiez un tableau de bord S3 Storage Lens, vous définissez la portée de ce tableau de bord et la sélection des métriques.

S3 Storage Lens fournit des métriques et des recommandations avancées, que vous pouvez utiliser moyennant des frais supplémentaires. Les métriques et recommandations avancées vous donnent accès à des métriques et à des fonctionnalités supplémentaires pour mieux comprendre

votre stockage. Ces fonctionnalités incluent des catégories de métriques avancées, l'agrégation de préfixes, des recommandations contextuelles et la publication sur Amazon CloudWatch. L'agrégation de préfixes et les recommandations contextuelles sont disponibles uniquement dans la console Amazon S3. Pour obtenir des informations sur la tarification de S3 Storage Lens, consultez la tarification Amazon S3.

Catégories de métriques

Aux niveaux gratuit et avancé, les métriques sont organisées en catégories correspondant aux principaux cas d'utilisation, tels que l'optimisation des coûts et la protection des données. Les métriques gratuites incluent le résumé, l'optimisation des coûts, la protection des données, la gestion des accès, les performances et les métriques d'événements. Lorsque vous passez à des métriques et à des recommandations avancées, vous pouvez activer des métriques avancées sur l'optimisation des coûts et la protection des données. Vous pouvez utiliser ces métriques avancées pour réduire encore vos coûts de stockage S3 et améliorer votre position en matière de protection des données. Vous pouvez également activer les métriques d'activité et les métriques de codes de statut détaillés afin d'améliorer les performances des charges de travail d'application qui accèdent à vos compartiments S3. Pour plus d'informations sur les catégories des métriques gratuites et avancées, consultez Sélection des métriques.

Vous pouvez évaluer votre stockage en fonction des bonnes pratiques dans S3, telles que l'analyse du pourcentage de compartiments pour lesquels le chiffrement, le verrouillage d'objets S3 ou la gestion des versions S3 sont activés. Vous pouvez également identifier les opportunités d'économies potentielles. Par exemple, vous pouvez utiliser des métriques relatives aux nombres de règles de cycle de vie S3 pour identifier les compartiments auxquels il manque des règles de cycle de vie d'expiration ou de transition. Vous pouvez également identifier analyser votre activité de demandes par compartiment afin de trouver des compartiments dont les objets pourraient être transférés vers une classe de stockage moins coûteuse. Pour de plus amples informations, veuillez consulter Cas d'utilisation des métriques Amazon S3 Storage Lens.

Exportation de métriques

Outre la visualisation du tableau de bord dans la console S3, vous pouvez exporter les métriques au format CSV ou Parquet vers un compartiment S3 pour une analyse approfondie avec l'outil d'analytique de votre choix. Pour de plus amples informations, veuillez consulter Afficher les métriques Amazon S3 Storage Lens à l'aide d'une exportation de données.

Publication sur Amazon CloudWatch

Vous pouvez publier des métriques d'utilisation et d'activité S3 Storage Lens sur Amazon CloudWatch pour créer une vue unifiée de l'état opérationnel dans les tableaux de bord CloudWatch. Vous pouvez également utiliser les fonctionnalités CloudWatch, comme les alarmes et les actions déclenchées, les mathématiques appliquées aux métriques et la détection d'anomalies pour surveiller les métriques S3 Storage Lens et agir dessus. En outre, les opérations d'API CloudWatch permettent aux applications, y compris à celles des fournisseurs tiers, d'accéder à vos métriques S3 Storage Lens. L'option de publication sur CloudWatch est disponible pour les tableaux de bord mis à niveau vers les métriques et recommandations avancées S3 Storage Lens. Pour plus d'informations sur la prise en charge des métriques S3 Storage Lens dans CloudWatch, consultez Surveillance des métriques S3 Storage Lens dans CloudWatch.

Pour plus d'informations sur l'utilisation de S3 Storage Lens, consultez les rubriques suivantes.

Rubriques

Présentation d'Amazon S3 Storage Lens

> **Important**
>
> Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. À partir du 5 janvier 2023, tous les nouveaux chargements d'objets sur Amazon S3 sont automatiquement chiffrés, sans coût supplémentaire et sans impact sur les performances. Le statut de chiffrement automatique pour la configuration de chiffrement par défaut du compartiment S3 et pour les nouveaux chargements d'objets est disponible dans les journaux AWS CloudTrail, S3 Inventory, S3 Storage Lens, dans la console Amazon S3, et en tant qu'en-tête de réponse supplémentaire de l'API Amazon S3 dans l'AWS Command Line

Interface et les kits SDK AWS. Pour plus d'informations, consultez la FAQ sur le chiffrement par défaut.

Amazon S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Vous pouvez utiliser les métriques S3 Storage Lens pour générer des informations récapitulatives, telles que la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes qui connaissent la croissance la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier les opportunités d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection et de sécurisation des données et améliorer les performances des charges de travail d'application. Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour faire expirer les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3. Vous pouvez créer et gérer des tableaux de bord S3 Storage Lens à l'aide de la console Amazon S3, d'AWS Command Line Interface (AWS CLI), des kits SDK AWS ou de l'API REST Amazon S3.

Concepts et terminologie propres à S3 Storage Lens

Cette section contient la terminologie et les concepts essentiels pour bien comprendre et utiliser Amazon S3 Storage Lens.

Rubriques

- Configuration du tableau de bord

- Tableau de bord par défaut
- Tableaux de bord
- Instantané du compte
- Exportation de métriques
- Région d'accueil
- Période de conservation
- Catégories de métriques
- Recommandations
- Sélection des métriques
- S3 Storage Lens et AWS Organizations

Configuration du tableau de bord

S3 Storage Lens nécessite une configuration de tableau de bord contenant les propriétés requises pour agréger les métriques en votre nom pour un seul tableau de bord ou une seule exportation. Lorsque vous créez une configuration, vous choisissez le nom du tableau de bord et la région d'origine, que vous ne pouvez pas modifier après avoir créé le tableau de bord. Vous pouvez éventuellement ajouter des étiquettes et configurer une exportation des métriques au format CSV ou Parquet.

Dans la configuration du tableau de bord, vous définissez également la portée du tableau de bord et la sélection des métriques. La portée peut inclure tout le stockage du compte de votre organisation ou des sections filtrées par région, compartiment et compte. Lorsque vous configurez la sélection de métriques, vous pouvez choisir entre des métriques gratuites et des métriques et recommandations avancées, que vous pouvez utiliser moyennant des frais supplémentaires. Les métriques et recommandations avancées vous donnent accès à des métriques et à des fonctionnalités supplémentaires. Ces fonctionnalités incluent des catégories de métriques avancées, l'agrégation au niveau du préfixe, des recommandations contextuelles et la publication sur Amazon CloudWatch. Pour obtenir des informations sur la tarification de S3 Storage Lens, consultez la tarification Amazon S3.

Tableau de bord par défaut

Le tableau de bord par défaut de S3 Storage Lens sur la console s'appelle default-account-dashboard. S3 préconfigure ce tableau de bord pour qu'il affiche le résumé des informations et des tendances pour l'ensemble de votre compte, et les met à jour quotidiennement dans la console S3.

Vous ne pouvez pas modifier la portée de la configuration du tableau de bord par défaut, mais vous pouvez mettre à niveau la sélection des métriques en passant des métriques gratuites aux métriques et recommandations avancées. Vous pouvez configurer l'exportation facultative des métriques ou même désactiver le tableau de bord. Toutefois, vous ne pouvez pas supprimer le tableau de bord par défaut.

> **Note**
>
> Si vous désactivez le tableau de bord par défaut, il n'est plus mis à jour. Vous ne recevrez plus de nouvelles métriques quotidiennes dans votre tableau de bord S3 Storage Lens, votre exportation de métriques ou l'instantané de compte sur la page S3 Compartiments. Si votre tableau de bord utilise des indicateurs et des recommandations avancés, vous ne serez plus facturé. Vous pourrez toujours voir les données d'historique dans le tableau de bord jusqu'à l'expiration du délai de 14 jours pour les requêtes de données. Cette période est de 15 mois si vous avez activé les métriques et recommandations avancées. Pour accéder aux données d'historique, vous pouvez réactiver le tableau de bord pendant la période d'expiration.

Tableaux de bord

Vous pouvez créer d'autres tableaux de bord S3 Storage Lens et les délimiter par Régions AWS, compartiments S3 ou comptes (pour AWS Organizations). Lorsque vous créez ou modifiez un tableau de bord S3 Storage Lens, vous définissez la portée de ce tableau de bord et la sélection des métriques. S3 Storage Lens fournit des métriques et des recommandations avancées, que vous pouvez utiliser moyennant des frais supplémentaires. Les métriques et recommandations avancées vous donnent accès à des métriques et à des fonctionnalités supplémentaires pour mieux comprendre votre stockage. Ces fonctionnalités incluent des catégories de métriques avancées, l'agrégation au niveau du préfixe, des recommandations contextuelles et la publication sur Amazon CloudWatch. Pour obtenir des informations sur la tarification de S3 Storage Lens, consultez la tarification Amazon S3.

Vous pouvez également désactiver ou supprimer les tableaux de bord. Si vous désactivez un tableau de bord, il ne sera plus mis à jour et vous ne recevrez plus de nouvelles métriques quotidiennes. Vous pouvez encore voir les données d'historique pendant la période d'expiration de 14 jours. Si vous avez activé les métriques et recommandations avancées pour ce tableau de bord, cette période est de 15 mois. Pour accéder aux données d'historique, vous pouvez réactiver le tableau de bord pendant la période d'expiration.

Si vous supprimez votre tableau de bord, vous perdrez tous ses paramètres de configuration. Vous ne recevrez plus de nouvelles métriques quotidiennes et vous perdrez également l'accès aux données historiques associées à ce tableau de bord. Si vous souhaitez accéder aux données historiques d'un tableau de bord supprimé, vous devez créer un autre tableau de bord portant le même nom dans la même Région d'accueil.

> **Note**
>
> - Vous pouvez utiliser S3 Storage Lens pour créer jusqu'à 50 tableaux de bord par Région d'accueil.
> - Les tableaux de bord au niveau de l'organisation ne peuvent être limités qu'à une portée régionale.

Instantané du compte

La fonctionnalité S3 Storage Lens Account snapshot (Instantané du compte) résume les métriques de votre tableau de bord par défaut et affiche votre stockage total, le nombre d'objets et la taille d'objet moyenne sur la page Buckets (compartiments) de la console S3. Cet instantané de compte vous permet d'accéder rapidement aux informations sur le stockage sans avoir à quitter la page Buckets (Compartiments). L'instantané du compte permet également d'accéder en un clic à votre tableau de bord S3 Storage Lens interactif.

Vous pouvez utiliser votre tableau de bord pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer des informations au niveau de l'organisation, du compte, du compartiment, de l'objet ou du préfixe. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

Vous ne pouvez pas modifier la portée du tableau de bord default-account dashboard (tableau de bord de compte par défaut) car il est lié à Account snapshot (Instantané du compte). Toutefois, vous pouvez mettre à niveau la sélection des métriques dans votre default-account-dashboard (tableau de bord de compte par défaut) en passant des métriques gratuites aux métriques et recommandations avancées payantes. Après cette mise à niveau, vous pouvez afficher toutes les demandes, les octets chargés et les octets téléchargés dans Account snapshot (Instantané du compte) S3 Storage Lens.

> **Note**
>
> Si vous désactivez votre tableau de bord par défaut, Account snapshot (Instantané du compte) n'est plus mis à jour. Pour continuer à afficher des métriques dans Account snapshot (Instantané du compte), vous pouvez réactiver default-account-dashboard (tableau de bord de compte par défaut).

Exportation de métriques

Une exportation de métriques S3 Storage Lens est un fichier contenant toutes les métriques identifiées dans votre configuration S3 Storage Lens. Ces informations sont générées quotidiennement au format CSV ou Parquet, et envoyées dans un compartiment S3. Vous pouvez utiliser l'exportation des métriques pour une analyse approfondie à l'aide de l'outil de métriques de votre choix. Le compartiment S3 pour l'exportation de vos métriques doit se trouver dans la même Région que la configuration de votre S3 Storage Lens. Vous pouvez générer une exportation de métriques S3 Storage Lens à partir de la console S3 en modifiant la configuration de votre tableau de bord. Vous pouvez également configurer une exportation de métriques en utilisant AWS CLI et les kits AWS SDK.

Région d'accueil

La région d'origine est la Région AWS dans laquelle toutes les métriques S3 Storage Lens d'une configuration de tableau de bord donnée sont stockées. Vous devez choisir une région d'origine lorsque vous créez votre configuration de tableau de bord S3 Storage Lens. Une fois que vous avez choisi une région d'origine, vous ne pouvez pas la modifier. De plus, si vous créez un groupe Storage Lens, nous vous recommandons de choisir la même région d'origine que celle de votre tableau de bord Storage Lens.

> **Note**
>
> Vous pouvez choisir l'une des régions suivantes comme région d'origine :
>
> - US East (N. Virginia) – `us-east-1`
> - US East (Ohio) – `us-east-2`
> - US West (N. California) – `us-west-1`
> - US West (Oregon) – `us-west-2`
> - Asia Pacific (Mumbai) – `ap-south-1`

- Asie-Pacifique (Séoul) – `ap-northeast-2`
- Asia Pacific (Singapore) – `ap-southeast-1`
- Asie-Pacifique (Sydney) – `ap-southeast-2`
- Asie-Pacifique (Tokyo) – `ap-northeast-1`
- Canada (Central) – `ca-central-1`
- Chine (Beijing) – `cn-north-1`
- Chine (Ningxia) – `cn-northwest-1`
- Europe (Frankfurt) – `eu-central-1`
- Europe (Ireland) – `eu-west-1`
- Europe (London) – `eu-west-2`
- Europe (Paris) – `eu-west-3`
- Europe (Stockholm) – `eu-north-1`
- South America (São Paulo) – `sa-east-1`

Période de conservation

Les métriques S3 Storage Lens sont conservées pour que vous puissiez voir les tendances historiques et comparer les différences de stockage et d'activité au fil du temps. Vous pouvez utiliser les métriques Amazon S3 Storage Lens pour les requêtes, pour observer les tendances historiques et comparer les différences d'utilisation et d'activité du stockage au fil du temps.

Toutes les métriques de S3 Storage Lens sont conservées pour une durée de 15 mois. Toutefois, les métriques ne sont disponibles que pour les requêtes d'une durée spécifique, qui dépend de votre sélection des métriques. Cette durée ne peut pas être modifiée. Pour les requêtes, les métriques gratuites sont disponibles pendant 14 jours et les métriques avancées sont disponibles pendant 15 mois.

Catégories de métriques

Dans les niveaux gratuit et avancé, les métriques S3 Storage Lens sont organisées en catégories correspondant aux principaux cas d'utilisation, tels que l'optimisation des coûts et la protection des données. Les métriques gratuites incluent le résumé, l'optimisation des coûts, la protection des données, la gestion des accès, les performances et les métriques d'événements. Lorsque vous passez à des métriques et à des recommandations avancées, vous pouvez activer des métriques supplémentaires sur l'optimisation des coûts et la protection des données que vous pouvez utiliser

pour réduire encore vos coûts de stockage S3 et assurer la protection de vos données. Vous pouvez également activer des métriques d'activité et des métriques de codes de statut détaillés que vous pouvez utiliser pour améliorer les performances des flux de travail des applications.

La liste suivante montre toutes les catégories de métriques gratuites et avancées. Pour obtenir la liste complète des métriques individuelles incluses dans chaque catégorie, consultez le glossaire des métriques.

Métriques de résumé

Les métriques récapitulatives fournissent des informations générales sur votre stockage S3, y compris le nombre total d'octets de stockage et le nombre d'objets.

Métriques sur l'optimisation des coûts

Les métriques sur l'optimisation des coûts fournissent des informations que vous pouvez utiliser pour gérer et optimiser vos coûts de stockage. Par exemple, vous pouvez identifier les compartiments contenant des chargements partitionnés non terminés datant de plus de 7 jours.

Les métriques et recommandations avancées vous permettent d'activer des métriques avancées sur l'optimisation des coûts. Ces métriques incluent les métriques relatives aux nombres de règles de cycle de vie S3 que vous pouvez utiliser pour obtenir les nombres de règles de cycle de vie S3 d'expiration et de transition par compartiment.

Métriques sur la protection des données

Les métriques sur la protection des données fournissent des informations sur les fonctionnalités de protection des données, telles que le chiffrement et la gestion des versions S3. Vous pouvez utiliser ces métriques pour identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données. Par exemple, vous pouvez identifier les compartiments qui n'utilisent pas le chiffrement par défaut avec les clés AWS Key Management Service (SSE-KMS) ou la gestion des versions S3.

Les métriques et recommandations avancées vous permettent d'activer des métriques avancées sur la protection des données. Ces métriques incluent des métriques relatives aux nombres de règles de réplication par compartiment.

Métriques de gestion des accès

Les métriques de gestion des accès fournissent des informations sur la propriété des objets S3. Vous pouvez utiliser ces métriques pour voir les paramètres de propriété d'objets utilisés par vos compartiments.

Métriques d'événements

Les métriques d'événements fournissent des informations pour les notifications d'événements S3. Les métriques d'événements vous permettent de voir quels compartiments ont des notifications d'événements S3 configurées.

Métriques de performances

Les métriques de performances fournissent des informations pour S3 Transfer Acceleration. Les métriques de performances vous permettent de voir dans quels compartiments l'accélération des transferts est activée.

Métriques d'activité (avancées)

Si vous mettez à niveau votre tableau de bord vers Métriques et recommandations avancées, vous pouvez activer les métriques d'activité. Les métriques d'activité fournissent des informations détaillées sur la façon dont votre stockage est sollicité par demandes (par exemple, Toutes les demandes, Demandes Get, Demandes Put), sur les octets chargés ou téléchargés et sur les erreurs.

Les métriques d'activité au niveau des préfixes peuvent être utilisées pour vous aider à déterminer quels préfixes sont rarement utilisés, afin que vous puissiez passer à une classe de stockage plus optimale à l'aide de S3 Lifecycle.

Métriques de codes de statut détaillées (avancées)

Si vous mettez à niveau votre tableau de bord vers Métriques et recommandations avancées, vous pouvez activer les métriques de codes de statut détaillées. Les métriques de codes de statut détaillées fournissent des informations sur les codes de statut HTTP, tels que 403 Interdit et 503 Service non disponible, que vous pouvez utiliser pour résoudre les problèmes d'accès ou de performances. Par exemple, vous pouvez consulter la métrique 403 Forbidden error count (Nombre d'erreurs 403 Interdit) pour identifier les charges de travail qui accèdent à des compartiments sans appliquer les autorisations appropriées.

Les métriques de codes de statut détaillées au niveau des préfixes peuvent être utilisées pour mieux comprendre les occurrences des codes de statut HTTP par préfixe. Par exemple, les métriques du nombre d'erreurs 503 vous permettent d'identifier les préfixes qui reçoivent des demandes de limitation lors de l'ingestion de données.

Recommandations

S3 Storage Lens fournit des recommandations automatiques pour vous aider à optimiser votre stockage. Les recommandations sont placées contextuellement à côté des métriques concernées

dans le tableau de bord S3 Storage Lens. Les données historiques ne sont pas éligibles aux recommandations, car les recommandations concernent ce qui se passe au cours de la période la plus récente. Les recommandations n'apparaissent que lorsqu'elles sont pertinentes.

Les recommandations de S3 Storage Lens sont présentées sous les formes suivantes :

- Suggestions

  Les suggestions vous alertent sur les tendances de stockage et d'activité pouvant indiquer une possibilité d'optimisation des coûts de stockage ou une bonne pratique de protection des données. Vous pouvez utiliser les rubriques suggérées dans le Guide de l'utilisateur Amazon S3 et le tableau de bord S3 Storage Lens pour effectuer une exploration permettant d'obtenir plus d'informations sur des régions, des compartiments ou des préfixes.
- Alertes

  Les alertes sont des recommandations qui vous signalent les anomalies intéressantes dans le stockage et l'activité au cours d'une période et qui pourraient nécessiter davantage d'attention ou de surveillance.
  - Alertes d'anomalies

    S3 Storage Lens fournit des alertes pour les métriques qui sont des anomalies, basées sur votre tendance des 30 derniers jours. Les anomalies sont calculées à l'aide d'un score standard, également connu sous le nom de score z. Dans ce score, la métrique du jour est soustraite de la moyenne des 30 derniers jours pour la métrique concernée. La métrique du jour est ensuite divisée par l'écart-type de cette métrique au cours des 30 derniers jours. Le score obtenu est généralement compris entre -3 et +3. Ce nombre représente le nombre d'écarts-types entre la métrique du jour et la moyenne.

    S3 Storage Lens considère les métriques ayant un score **>** 2 ou **<** -2 comme des anomalies, car elles sont supérieures ou inférieures à 95 % des données normalement distribuées.
  - Alertes de changement significatif

    L'alerte de changement significatif s'applique aux métriques qui sont censées changer moins fréquemment. Par conséquent, elle est définie sur une sensibilité plus élevée que le calcul des anomalies, qui se situe généralement dans un intervalle de +/- 20 % par rapport au jour, à la semaine ou au mois précédents.

    Réagir aux alertes liées au stockage et à l'activité : une alerte de changement significatif ne reflète pas nécessairement un problème. Elle peut être le résultat d'un changement prévu

de votre stockage. Par exemple, il se peut que vous ayez récemment ajouté ou supprimé un nombre important d'objets ou apporté d'autres modifications planifiées de ce type.

Si vous voyez une alerte de changement significatif sur votre tableau de bord, prenez-en note et déterminez si elle peut être expliquée par des circonstances récentes. Si ce n'est pas le cas, utilisez le tableau de bord S3 Storage Lens pour effectuer une exploration et obtenir plus de détails afin de comprendre quels Régions, compartiments ou préfixes spécifiques sont à l'origine de la fluctuation.

- Rappels

  Les rappels fournissent des informations sur le fonctionnement d'Amazon S3. Ils peuvent vous aider à comprendre comment utiliser les fonctions S3 permettant de réduire les coûts de stockage ou d'appliquer les bonnes pratiques de protection des données.

Sélection des métriques

S3 Storage Lens propose deux sélections de métriques que vous pouvez choisir pour votre tableau de bord et pour l'exportation : des métriques gratuites et des métriques et recommandations avancées.

- Métriques gratuites

  S3 Storage Lens propose des métriques gratuites pour tous les tableaux de bord et configurations. Les métriques gratuites contiennent des métriques pertinentes pour votre stockage, telles que le nombre de compartiments et les objets dans votre compte. Les métriques gratuites incluent également des métriques basées sur des cas d'utilisation (par exemple, les métriques sur l'optimisation des coûts et la protection des données) que vous pouvez utiliser pour vérifier si votre stockage est configuré conformément aux bonnes pratiques dans S3. L'ensemble des métriques gratuites sont collectées quotidiennement. Les données sont disponibles pour les requêtes pendant 14 jours. Pour plus d'informations sur les métriques disponibles avec les métriques gratuites, consultez Glossaire des métriques Amazon S3 Storage Lens.

- Métriques et recommandations avancées

  S3 Storage Lens offre des métriques gratuites pour tous les tableaux de bord et configurations, avec la possibilité d'une mise à niveau vers les métriques et recommandations avancées. Des frais supplémentaires seront facturés. Pour de plus amples informations, veuillez consulter Tarification Amazon S3.

Les métriques et recommandations avancées incluent toutes les métriques gratuites et des métriques supplémentaires, telles que les métriques avancées sur la protection des données et l'optimisation des coûts, ainsi que des métriques d'activité et des métriques de codes de statut détaillés. Les métriques et recommandations avancées fournissent également des recommandations pour vous aider à optimiser votre stockage. Les recommandations sont placées contextuellement à côté des métriques concernées dans le tableau de bord.

Les métriques et recommandations avancées incluent les fonctionnalités suivantes :

- Métriques avancées : générez des métriques supplémentaires. Pour obtenir la liste complète des catégories de métriques avancées, consultez Catégories de métriques. Pour obtenir la liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.
- Publication sur Amazon CloudWatch : publie des métriques S3 Storage Lens sur CloudWatch pour créer une vue unifiée de l'état opérationnel dans les tableaux de bord CloudWatch. Vous pouvez également utiliser les fonctionnalités et opérations d'API CloudWatch, comme les alarmes et les actions déclenchées, les mathématiques appliquées aux métriques et la détection d'anomalies pour surveiller les métriques S3 Storage Lens et agir dessus. Pour de plus amples informations, veuillez consulter Surveillance des métriques S3 Storage Lens dans CloudWatch.
- Regroupement des préfixes : collecte des métriques au niveau du préfixe. L'activation du regroupement des préfixes étend toutes les métriques incluses dans la configuration de votre tableau de bord au niveau du préfixe. Les métriques ne sont générées que pour les préfixes qui atteignent le seuil configuré. Notez que les métriques applicables au niveau du préfixe sont disponibles avec la fonctionnalité Regroupement des préfixes, à l'exception des paramètres au niveau du compartiment et des métriques relatives au nombre de règles. Les métriques au niveau du préfixe ne sont pas publiées dans CloudWatch.
- Regroupement des groupes Storage Lens : collecte des métriques au niveau du groupe Storage Lens. Après avoir activé Métriques et recommandations avancées et Regroupement des groupes Storage Lens, vous pouvez spécifier les groupes Storage Lens à inclure ou à exclure de votre tableau de bord Storage Lens. Au moins un groupe Storage Lens doit être spécifié. Les groupes Storage Lens spécifiés doivent également résider dans la région d'origine désignée dans le compte du tableau de bord. Les métriques au niveau du groupe Storage Lens ne sont pas publiées dans CloudWatch.

L'ensemble des métriques avancées sont collectées quotidiennement. Les données sont disponibles pour les requêtes jusqu'à 15 mois. Pour plus d'informations sur les métriques de stockage agrégées par S3 Storage Lens, consultez Glossaire des métriques Amazon S3 Storage Lens.

> **Note**
>
> Les recommandations sont disponibles uniquement lorsque vous utilisez le tableau de bord S3 Storage Lens sur la console Amazon S3.

S3 Storage Lens et AWS Organizations

AWS Organizations est un Service AWS qui vous aide à regrouper tous vos Comptes AWS dans une hiérarchie d'organisation unique. Combiné à AWS Organizations, Amazon S3 Storage Lens fournit une vue unique du stockage et de l'activité des objets sur votre stockage Amazon S3.

Pour de plus amples informations, veuillez consulter Utilisation d'Amazon S3 Storage Lens avec AWS Organizations.

- Accès sécurisé

  À l'aide du compte de gestion de votre organisation, vous devez activer l'accès sécurisé pour S3 Storage Lens afin d'agréger les métriques de stockage et les données d'utilisation pour tous les comptes membres de votre organisation. Vous pouvez ensuite créer des tableaux de bord ou des exportations pour votre organisation à l'aide de votre compte de gestion ou en accordant un accès administrateur délégué à d'autres comptes de votre organisation.

  Vous pouvez désactiver l'accès sécurisé de S3 Storage Lens à tout moment pour empêcher S3 Storage Lens d'agréger les métriques de votre organisation.

- Administrateur délégué

  Vous pouvez créer des tableaux de bord et des métriques pour S3 Storage Lens pour votre organisation à l'aide de votre compte de gestion AWS Organizations ou en accordant un accès administrateur délégué à d'autres comptes de votre organisation. Vous pouvez désinscrire les administrateurs délégués à tout moment. La désinscription d'un administrateur délégué empêche également automatiquement tous les tableaux de bord au niveau de l'organisation créés par cet administrateur délégué d'agréger de nouvelles métriques de stockage.

Pour de plus amples informations, veuillez consulter la section Amazon S3 Storage Lens et AWS Organizations du Guide de l'utilisateur AWS Organizations.

Rôles liés à un service Amazon S3 Storage Lens

Outre l'accès sécurisé d'AWS Organizations, Amazon S3 Storage Lens utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement au S3 Storage Lens. Les rôles liés à un service sont prédéfinis par S3 Storage Lens et incluent toutes les autorisations nécessaires pour collecter les métriques de stockage et d'activité quotidiennes des comptes membres de votre organisation.

Pour plus d'informations, consultez la section Utilisation des rôles liés à un service pour Amazon S3 Storage Lens.

Utilisation d'Amazon S3 Storage Lens avec AWS Organizations

Amazon S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Vous pouvez utiliser les métriques S3 Storage Lens pour générer des informations récapitulatives, telles que la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes qui connaissent la croissance la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier les opportunités d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection et de sécurisation des données et améliorer les performances des charges de travail d'application. Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour faire expirer les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

Vous pouvez utiliser Amazon S3 Storage Lens pour collecter des métriques de stockage et des données d'utilisation pour tous les Comptes AWS qui font partie de votre hiérarchie AWS Organizations. Pour ce faire, vous devez utiliser AWS Organizations et activer l'accès approuvé de S3 Storage Lens à l'aide de votre compte de gestion AWS Organizations.

Une fois l'accès sécurisé activé, vous pouvez ajouter un accès administrateur délégué aux comptes de votre organisation. Ces comptes peuvent ensuite créer des configurations et des tableaux de bord S3 Storage Lens qui collectent des métriques de stockage et des données utilisateur à l'échelle de l'organisation.

Pour de plus informations sur l'activation d'un accès sécurisé, veuillez consulter la section Amazon S3 Storage Lens et AWS Organizations du Guide de l'utilisateur AWS Organizations.

Rubriques

Activer l'accès sécurisé pour S3 Storage Lens

En activant l'accès approuvé, vous autorisez Amazon S3 Storage Lens à accéder à vos hiérarchie, adhésion et structure AWS Organizations via les opérations d'API AWS Organizations. S3 Storage Lens devient alors un service sécurisé pour l'ensemble de la structure de votre organisation.

Chaque fois qu'une configuration de tableau de bord est créée, S3 Storage Lens crée des rôles liés à un service dans les comptes de gestion ou d'administrateur délégué de votre organisation. Le rôle lié à un service accorde à S3 Storage Lens l'autorisation d'effectuer les opérations suivantes :

- Décrire les organisations
- Répertorier les comptes
- Vérifier une liste d'accès au Service AWS pour les organisations
- Obtenir des administrateurs délégués pour les organisations

S3 Storage Lens peut alors s'assurer qu'il dispose d'un accès pour collecter les métriques entre comptes pour les comptes de vos organisations. Pour plus d'informations, consultez la section Utilisation des rôles liés à un service pour Amazon S3 Storage Lens.

Une fois l'accès approuvé activé, vous pouvez affecter un accès administrateur délégué aux comptes de votre organisation. Lorsqu'un compte est marqué en tant qu'administrateur délégué pour un service, le compte reçoit l'autorisation d'accéder à toutes les opérations d'API d'organisation en lecture seule. Cet accès fournit une visibilité d'administrateur délégué aux membres et aux structures de votre organisation pour leur permettre à eux aussi de créer des tableaux de bord S3 Storage Lens.

> **Note**
>
> Seul le compte de gestion peut activer un accès sécurisé pour Amazon S3 Storage Lens.

Désactiver l'accès sécurisé pour S3 Storage Lens

En désactivant l'accès sécurisé, vous limitez le fonctionnement de S3 Storage Lens au niveau du compte uniquement. En outre, chaque titulaire de compte peut voir uniquement les informations S3 Storage Lens correspondant à la portée de son compte, et non pas à l'ensemble de l'organisation. Les tableaux de bord qui nécessitent un accès approuvé ne sont plus mis à jour, mais conservent leurs données d'historique pour la période de disponibilité des données pour les requêtes.

> **Note**
>
> - La désactivation de l'accès approuvé pour S3 Storage Lens empêche également automatiquement tous les tableaux de bord au niveau de l'organisation de collecter et d'agréger des métriques de stockage.
> - Vos comptes de gestion et d'administrateur délégué peuvent toujours voir les données d'historique pour vos tableaux de bord au niveau de votre organisation au cours de la période de disponibilité des données pour les requêtes.

Enregistrer un administrateur délégué S3 Storage Lens

Vous pouvez créer des tableaux de bord au niveau de l'organisation à l'aide des comptes de gestion ou d'administrateur délégué de votre organisation. Les comptes d'administrateur délégué permettent à d'autres comptes que votre compte de gestion de créer des tableaux de bord au niveau de l'organisation. Seul le compte de gestion d'une organisation peut enregistrer et annuler d'autres comptes en tant qu'administrateurs délégués pour l'organisation.

Pour enregistrer un administrateur délégué à l'aide de la console Amazon S3, consultez Enregistrer des administrateurs délégués pour S3 Storage Lens.

Vous pouvez également enregistrer un administrateur délégué à l'aide de l'API REST AWS Organizations, d'AWS CLI ou des kits SDK à partir du compte de gestion. Pour plus d'informations, consultez RegisterDelegatedAdministrator dans la Référence d'API AWS Organizations.

> **Note**
>
> Avant de pouvoir désigner un administrateur délégué à l'aide de l'API REST AWS Organizations, d'AWS CLI ou des kits SDK, vous devez appeler l'opération EnableAWSOrganizationsAccess.

Annuler l'enregistrement d'un administrateur délégué pour S3 Storage Lens

Vous pouvez également désinscrire un compte d'administrateur délégué. Les comptes d'administrateur délégué permettent à d'autres comptes que votre compte de gestion de créer des tableaux de bord au niveau de l'organisation. Seul le compte de gestion d'une organisation peut désinscrire des comptes en tant qu'administrateurs délégués pour l'organisation.

Pour désinscrire un administrateur délégué à l'aide de la console S3, consultez Annuler l'enregistrement d'administrateurs délégués pour S3 Storage Lens.

Vous pouvez également désinscrire un administrateur délégué à l'aide de l'API REST AWS Organizations, d'AWS CLI ou des kits SDK à partir du compte de gestion. Pour plus d'informations, consultez DeregisterDelegatedAdministrator dans la Référence d'API AWS Organizations.

> **Note**
>
> - La désinscription d'un administrateur délégué empêche également automatiquement tous les tableaux de bord au niveau de l'organisation créés par cet administrateur délégué d'agréger de nouvelles métriques de stockage.
> - L'administrateur délégué désinscrit peut toujours consulter les données d'historique des tableaux de bord qu'ils ont créés tant que les données sont disponibles pour les requêtes.

Autorisations Amazon S3 Storage Lens

Amazon S3 Storage Lens nécessite de nouvelles autorisations dans AWS Identity and Access Management (IAM) pour autoriser l'accès aux actions S3 Storage Lens. Pour accorder ces autorisations, vous pouvez utiliser une politique IAM basée sur l'identité. Vous pouvez attacher cette politique aux utilisateurs, groupes ou rôles IAM pour leur accorder des autorisations. Ces autorisations peuvent inclure la possibilité d'activer ou de désactiver S3 Storage Lens ou d'accéder à n'importe quel tableau de bord ou n'importe quelle configuration S3 Storage Lens.

L'utilisateur ou le rôle IAM doit appartenir au compte qui a créé ou qui détient le tableau de bord ou la configuration, sauf si les deux conditions suivantes sont remplies :

- Votre compte est membre d'AWS Organizations.
- Vous avez été autorisé à créer des tableaux de bord au niveau de l'organisation par votre compte de gestion en tant qu'administrateur délégué.

> **Note**
>
> - Vous ne pouvez pas utiliser les informations d'identification de l'utilisateur root de votre compte pour afficher les tableaux de bord Amazon S3 Storage Lens. Pour accéder aux tableaux de bord S3 Storage Lens, vous devez accorder les autorisations IAM requises à un utilisateur IAM nouveau ou existant. Connectez-vous ensuite à l'aide de ces informations d'identification utilisateur pour accéder aux tableaux de bord S3 Storage Lens. Pour plus d'informations, consultez la rubrique Bonnes pratiques IAM du Guide de l'utilisateur IAM.
> - L'utilisation de S3 Storage Lens sur la console Amazon S3 peut nécessiter plusieurs autorisations. Par exemple, pour modifier un tableau de bord sur la console, vous devez disposer des autorisations suivantes :
>   - `s3:ListStorageLensConfigurations`
>   - `s3:GetStorageLensConfiguration`
>   - `s3:PutStorageLensConfiguration`

Rubriques

Définir des autorisations de compte pour utiliser S3 Storage Lens

Pour créer et gérer des tableaux de bord S3 Storage Lens et des configurations de tableau de bord Storage Lens, vous devez disposer des autorisations suivantes, en fonction des opérations que vous souhaitez effectuer :

Autorisations IAM liées à Amazon S3 Storage Lens

Action	Autorisations IAM
Créez ou mettez à jour un tableau de bord S3 Storage Lens dans la console Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:GetStorageLensConfigurationTagging` `s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Obtenez les étiquettes d'un tableau de bord S3 Storage Lens sur la console Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfigurationTagging`
Affichez un tableau de bord S3 Storage Lens sur la console Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:GetStorageLensDashboard`
Supprimez un tableau de bord S3 Storage Lens sur la console Amazon S3.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration` `s3:DeleteStorageLensConfiguration`
Créez ou mettez à jour une configuration S3 Storage Lens en utilisant AWS CLI ou un kit AWS SDK.	`s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Obtenez les étiquettes d'une configuration S3 Storage Lens en utilisant AWS CLI ou un kit AWS SDK.	`s3:GetStorageLensConfigurationTagging`

Action	Autorisations IAM
Affichez une configuration S3 Storage Lens en utilisant AWS CLI ou un kit AWS SDK.	`s3:GetStorageLensConfiguration`
Supprimez une configuration S3 Storage Lens en utilisant AWS CLI ou un kit AWS SDK.	`s3:DeleteStorageLensConfiguration`

> **Note**
>
> - Vous pouvez utiliser des balises de ressources dans une stratégie IAM pour gérer les autorisations.
> - Un rôle ou un utilisateur IAM disposant de ces autorisations peut voir des métriques à partir de compartiments et de préfixes pour lesquels il peut ne pas avoir d'autorisation directe pour lire ou répertorier les objets.
> - Pour les tableaux de bord S3 Storage Lens sur lesquels les métriques au niveau du préfixe sont activées, si le chemin d'un préfixe sélectionné correspond à une clé d'objet, le tableau de bord peut afficher la clé d'objet sous la forme d'un autre préfixe.
> - Pour les exportations de métriques, qui sont stockées dans un compartiment de votre compte, les autorisations sont accordées à l'aide de l'autorisation `s3:GetObject` existante dans la politique IAM. De même, pour une entité AWS Organizations, le compte de gestion ou les comptes d'administrateur délégué de l'organisation peuvent utiliser des politiques IAM pour gérer les autorisations d'accès pour le tableau de bord et les configurations au niveau de l'organisation.

Définition d'autorisations de compte pour utiliser des groupes S3 Storage Lens

Vous pouvez utiliser les groupes S3 Storage Lens pour comprendre la distribution de votre espace de stockage au sein des compartiments en fonction du préfixe, du suffixe, de la balise d'objet, de la taille de l'objet ou de l'âge de l'objet. Vous pouvez attacher des groupes Storage Lens à vos tableaux de bord pour consulter leurs métriques agrégées.

Pour utiliser des groupes Storage Lens, vous devez disposer de certaines autorisations. Pour de plus amples informations, veuillez consulter the section called “Autorisations pour les groupes Storage Lens”.

Définir des autorisations pour S3 Storage Lens à l'aide d AWS Organizations

Vous pouvez utiliser Amazon S3 Storage Lens pour collecter des métriques de stockage et des données d'utilisation pour tous les comptes qui font partie de votre hiérarchie AWS Organizations. Voici les actions et autorisations liées à l'utilisation de S3 Storage Lens avec Organizations.

AWS OrganizationsAutorisations IAM liées à pour l'utilisation de S3 Storage Lens

Action	Autorisations IAM
Activez l'accès sécurisé de S3 Storage Lens à votre organisation.	`organizations:EnableAWSServiceAccess`
Désactivez l'accès approuvé pour S3 Storage Lens pour votre organisation.	`organizations:DisableAWSServiceAccess`
Enregistrez un administrateur délégué pour créer des tableaux de bord ou des configurations S3 Storage Lens pour votre organisation.	`organizations:RegisterDelegatedAdministrator`
Désinscrivez un administrateur délégué afin qu'il ne puisse plus créer de tableaux de bord ni de configurations S3 Storage Lens pour votre organisation.	`organizations:DeregisterDelegatedAdministrator`
Autorisations supplémentaires pour créer des configurations S3 Storage Lens à l'échelle de l'organisation	`organizations:DescribeOrganization` `organizations:ListAccounts` `organizations:ListAWSServiceAccessForOrganization` `organizations:ListDelegatedAdministrators` `iam:CreateServiceLinkedRole`

Affichage des métriques avec Amazon S3 Storage Lens

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

Par défaut, tous les tableaux de bord sont configurés avec les métriques gratuites, qui incluent les métriques que vous pouvez utiliser pour comprendre l'utilisation et l'activité sur votre espace de stockage S3, optimiser vos coûts de stockage et mettre en œuvre les bonnes pratiques de protection des données et de gestion des accès. Les métriques gratuites sont agrégées jusqu'au niveau du compartiment. Avec les métriques gratuites, les données sont disponibles pour les requêtes jusqu'à 14 jours.

Les métriques et recommandations avancées incluent les fonctionnalités supplémentaires suivantes que vous pouvez utiliser pour mieux comprendre l'utilisation et l'activité de votre stockage, ainsi que les bonnes pratiques visant à optimiser votre stockage :

- Recommandations contextuelles (disponibles uniquement dans le tableau de bord)
- Métriques avancées (y compris les métriques d'activité agrégées par compartiment)
- Prefix aggregation (Agrégation de préfixes)
- Regroupement des groupes Storage Lens
- Regroupement des groupes Storage Lens
- Publication sur Amazon CloudWatch

Les métriques avancées sont disponibles pour les requêtes pendant 15 mois. L'utilisation de S3 Storage Lens avec des métriques avancées entraîne des frais supplémentaires. Pour de plus amples informations, veuillez consulter Tarification Amazon S3. Pour plus d'informations sur les métriques gratuites et avancées, consultez Sélection des métriques.

Rubriques

- Afficher les métriques S3 Storage Lens sur les tableaux de bord

- Afficher les métriques Amazon S3 Storage Lens à l'aide d'une exportation de données
- Surveillance des métriques S3 Storage Lens dans CloudWatch

Afficher les métriques S3 Storage Lens sur les tableaux de bord

Dans la console Amazon S3, S3 Storage Lens fournit un tableau de bord interactif par défaut que vous pouvez utiliser pour visualiser les informations et les tendances de vos données. Vous pouvez également utiliser ce tableau de bord pour signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer des informations au niveau du compte, du compartiment, de la Région AWS, du préfixe ou du groupe Storage Lens. Si vous avez fait en sorte que S3 Storage Lens fonctionne avec AWS Organizations, vous pouvez également générer des informations au niveau de l'organisation (telles que des données pour tous les comptes qui font partie de votre hiérarchie AWS Organizations). Le tableau de bord se charge toujours à la date la plus récente pour laquelle des métriques sont disponibles.

Le tableau de bord par défaut de S3 Storage Lens sur la console s'appelle default-account-dashboard. Amazon S3 préconfigure ce tableau de bord pour qu'il affiche le résumé des informations et des tendances pour l'ensemble de votre compte, et les met à jour quotidiennement dans la console S3. Vous ne pouvez pas modifier la portée de la configuration du tableau de bord par défaut, mais vous pouvez mettre à niveau la sélection en passant des métriques gratuites aux métriques et recommandations avancées payantes. Les métriques et recommandations avancées vous donnent accès à des métriques et à des fonctionnalités supplémentaires. Ces fonctionnalités incluent des catégories de métriques avancées, l'agrégation au niveau du préfixe, des recommandations contextuelles et la publication sur Amazon CloudWatch.

Vous pouvez désactiver le tableau de bord par défaut, mais pas le supprimer. Si vous désactivez le tableau de bord par défaut, il n'est plus mis à jour. De plus, vous ne recevrez plus de nouvelles métriques quotidiennes dans S3 Storage Lens ou dans la section Instantané de compte sur la page Compartiments. Vous pourrez toujours voir les données d'historique dans le tableau de bord par défaut jusqu'à l'expiration du délai de 14 jours pour les requêtes de données. Cette période est de 15 mois si vous avez activé les métriques et recommandations avancées. Pour accéder à ces données, vous pouvez réactiver le tableau de bord par défaut pendant la période d'expiration.

Vous pouvez créer d'autres tableaux de bord S3 Storage Lens et les délimiter par Régions AWS, compartiments S3 ou comptes. Vous pouvez également définir vos tableaux de bord par organisation si vous avez fait en sorte que Storage Lens fonctionne avec AWS Organizations. Lorsque vous créez

ou modifiez un tableau de bord S3 Storage Lens, vous définissez la portée de ce tableau de bord et la sélection des métriques.

Vous pouvez désactiver ou supprimer tous les tableaux de bord supplémentaires que vous créez.

- Si vous désactivez un tableau de bord, il ne sera plus mis à jour et vous ne recevrez plus de nouvelles métriques quotidiennes. Vous pouvez encore voir les données d'historique des métriques gratuites pendant la période d'expiration de 14 jours. Si vous avez activé les métriques et recommandations avancées pour ce tableau de bord, cette période est de 15 mois. Pour accéder à ces données, vous pouvez réactiver le tableau de bord pendant la période d'expiration.
- Si vous supprimez votre tableau de bord, vous perdrez tous ses paramètres de configuration. Vous ne recevrez plus de nouvelles métriques quotidiennes et vous perdrez également l'accès aux données historiques associées à ce tableau de bord. Si vous souhaitez accéder aux données historiques d'un tableau de bord supprimé, vous devez créer un autre tableau de bord portant le même nom dans la même Région d'accueil.

Rubriques

Afficher un tableau de bord Amazon S3 Storage Lens

La procédure suivante montre comment afficher un tableau de bord S3 Storage Lens dans la console S3. Pour accéder à des procédures pas à pas basées sur des cas d'utilisation qui montrent comment utiliser votre tableau de bord pour optimiser les coûts, mettre en œuvre les bonnes pratiques et améliorer les performances des applications qui accèdent à vos compartiments S3, consultez Cas d'utilisation des métriques Amazon S3 Storage Lens.

> **Note**
>
> Vous ne pouvez pas utiliser les informations d'identification de l'utilisateur root de votre compte pour afficher les tableaux de bord Amazon S3 Storage Lens. Pour accéder aux tableaux de bord S3 Storage Lens, vous devez accorder les autorisations AWS Identity and Access Management (IAM) requises à un utilisateur IAM nouveau ou existant. Connectez-vous ensuite à l'aide de ces informations d'identification utilisateur pour accéder aux tableaux de bord S3 Storage Lens. Pour en savoir plus, consultez Autorisations Amazon

S3 Storage Lens et les Bonnes pratiques de sécurité dans IAM que vous trouverez dans le guide de l'utilisateur IAM.

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.

   Votre tableau de bord s'ouvre dans S3 Storage Lens. La section Snapshot for date (Instantané pour date) affiche la date la plus récente pour laquelle S3 Storage Lens a collecté des métriques. Votre tableau de bord charge toujours la date la plus récente pour laquelle des métriques sont disponibles.
4. (Facultatif) Pour modifier la date de votre tableau de bord S3 Storage Lens, choisissez une nouvelle date dans le sélecteur de date en haut à droite.
5. (Facultatif) Pour appliquer des filtres temporaires afin de limiter davantage la portée de vos données de tableau de bord, procédez comme suit :

   a. Développez la section Filtres.

   b. Pour filtrer par comptes, Régions AWS, classes de stockage, compartiments, préfixes ou groupes Storage Lens spécifiques, choisissez les options selon lesquelles vous souhaitez filtrer.

      > **Note**
      >
      > Le filtre Préfixes et le filtre Groupes Storage Lens ne peuvent pas être appliqués en même temps.

   c. Pour mettre à jour un filtre, choisissez Apply (Appliquer).

   d. Pour supprimer un filtre, cliquez sur X en regard du filtre.
6. Dans n'importe quelle section de votre tableau de bord S3 Storage Lens, pour voir les données relatives à une métrique spécifique, dans Metric (Métrique), choisissez le nom de la métrique.
7. Dans un graphique ou une visualisation quelconque de votre tableau de bord S3 Storage Lens, vous pouvez explorer des niveaux d'agrégation plus profonds à l'aide des onglets Comptes,

Régions AWS, Classes de stockage, Compartiments, Préfixes ou Groupes Storage Lens. Pour voir un exemple, consultez Découverte des compartiments Amazon S3 froids.

Comprendre votre tableau de bord S3 Storage Lens

Votre tableau de bord S3 Storage Lens comprend un onglet Overview (Présentation) principal et jusqu'à cinq onglets supplémentaires représentant les différents niveaux d'agrégation :

- Comptes
- Régions AWS
- Classes de stockage
- Compartiments
- Préfixes
- Groupes Storage Lens

Dans l'onglet Overview (Présentation), vos données de tableau de bord sont regroupées en trois sections différentes : Snapshot for date (Instantané pour date), Trends and distributions (Tendances et distributions) et Top N overview (Aperçu des N éléments principaux).

Pour plus d'informations sur votre tableau de bord S3 Storage Lens, consultez les sections suivantes.

Instantané

La section Snapshot for date (Instantané pour date) affiche les métriques récapitulatives que S3 Storage Lens a agrégées pour la date sélectionnée. Ces métriques récapitulatives incluent les métriques suivantes :

- Stockage total : quantité totale de stockage utilisée en octets.
- Nombre d'objets : nombre total d'objets dans votre Compte AWS.
- Taille moyenne d'objet : taille moyenne des objets.
- Compartiments actifs : nombre total de compartiments activement utilisés dont le stockage est supérieur à 0 octet dans votre compte.
- Comptes : nombre de comptes dont le stockage est dans la portée. Cette valeur est définie sur 1, sauf si vous utilisez AWS Organizations et que S3 Storage Lens dispose d'un accès approuvé avec un rôle lié à un service valide. Pour de plus amples informations, veuillez consulter Utilisation des rôles liés à un service pour le cadre de stockage Amazon S3.

- Compartiments : nombre total de compartiments dans votre compte.

Données des métriques

Pour chaque métrique qui apparaît dans l'instantané, vous pouvez voir les données suivantes :

- Nom de la métrique : nom de la métrique.
- Catégorie de métrique : catégorie dans laquelle la métrique est organisée.
- Total pour date : nombre total pour la date sélectionnée.
- % de variation : pourcentage de variation par rapport à la date du dernier instantané.
- Tendance sur 30 jours : courbe de tendance montrant les variations de la métrique sur une période de 30 jours.
- Recommandation : recommandation contextuelle basée sur les données fournies dans l'instantané. Les recommandations sont disponibles avec les métriques et recommandations avancées. Pour de plus amples informations, veuillez consulter Recommandations.

Catégories de métriques

Vous pouvez éventuellement mettre à jour la section Snapshot for date (Instantané pour date) de votre tableau de bord afin d'afficher les métriques relatives à d'autres catégories. Si vous souhaitez voir des données d'instantané pour des métriques supplémentaires, vous pouvez choisir parmi les Metrics categories (Catégories de métriques) suivantes :

- Optimisation des coûts
- Protection des données
- Activité (disponible avec les métriques avancées)
- Gestion des accès
- Performances
- Evénements

La section Snapshot for date (Instantané pour date) affiche uniquement une sélection de métriques pour chaque catégorie. Pour voir toutes les métriques d'une catégorie spécifique, choisissez la métrique dans les sections Trends and distributions (Tendances et distributions) ou Top N overview (Aperçu des N éléments principaux). Pour plus d'informations sur les catégories de métriques,

consultez Catégories de métriques. Pour obtenir la liste complète des métriques S3 Storage Lens, consultez Glossaire des métriques Amazon S3 Storage Lens.

Tendances et distributions

La deuxième section de l'onglet Overview (Présentation) est Trends and distributions (Tendances et distributions). Dans la section Trends and distributions (Tendances et distributions), vous pouvez choisir deux métriques à comparer sur une plage de dates que vous définissez. La section Trends and distributions (Tendances et distributions) montre la relation entre deux métriques au fil du temps. Cette section contient des graphiques montrant la distribution Storage class (Classe de stockage) et Region (Région) entre les deux tendances que vous suivez. Vous pouvez éventuellement explorer un point de données dans l'un des graphiques pour une analyse approfondie.

Pour une procédure pas à pas utilisant la section Trends and distributions (Tendances et distributions), consultez Identification des compartiments qui n'utilisent pas le chiffrement côté serveur avec AWS KMS pour le chiffrement par défaut (SSE-KMS).

Aperçu des N éléments principaux

La troisième section du tableau de bord S3 Storage Lens est Top N overview (Aperçu des N éléments principaux) (triés par ordre croissant ou décroissant). Cette section affiche vos métriques sélectionnées sur le plus grand nombre comptes, de Régions AWS, de compartiments, de préfixes ou de groupes Storage Lens. Si vous avez fait en sorte que S3 Storage Lens fonctionne avec AWS Organizations, vous pouvez également voir vos métriques sélectionnées dans votre organisation.

Pour une procédure pas à pas utilisant la section Top N overview (Aperçu des N éléments principaux), consultez Identifier vos compartiments S3 les plus importants.

Exploration et analyse par options

Pour vous offrir une expérience d'analyse fluide, le tableau de bord S3 Storage Lens fournit un menu d'actions qui s'affiche lorsque vous choisissez une valeur de graphique quelconque. Pour utiliser ce menu, choisissez une valeur de graphique quelconque pour afficher les valeurs de métriques associées, puis choisissez l'une des deux options suivantes dans la zone qui apparaît :

- L'action Drill down (Explorer) applique la valeur sélectionnée en tant que filtre dans tous les onglets de votre tableau de bord. Vous pouvez ensuite explorer cette valeur pour une analyse plus approfondie.
- L'action Analyser par vous redirige vers l'onglet Dimension que vous sélectionnez et applique cette valeur d'onglet en tant que filtre. Ces onglets incluent Comptes, Régions AWS, Classes

de stockage, Compartiments, Préfixes (pour les tableaux de bord où Métriques avancées et Regroupement des préfixes sont activés) et Groupes Storage Lens (pour les tableaux de bord où Métriques avancées et Regroupement des groupes Storage Lens sont activés). Analyser par vous permet de visualiser les données dans le contexte de la nouvelle dimension pour une analyse approfondie.

Les actions Approfondir et Analyser par peuvent être désactivées en cas de résultats illogiques ou dépourvus de valeur. Les actions Approfondir et Analyser par appliquent des filtres en plus de tous les filtres existants dans tous les onglets du tableau de bord. Vous pouvez également supprimer les filtres si nécessaire.

Onglets

Les onglets de niveau de dimension fournissent une vue détaillée de toutes les valeurs au sein d'une dimension particulière. Par exemple, l'onglet Régions AWS affiche les métriques de toutes les Régions AWS et l'onglet Compartiment affiche les métriques de tous les compartiments. Chaque onglet de dimension contient une mise en page identique composée de quatre sections :

- Un graphique de tendances qui affiche les N premiers éléments de la dimension au cours des 30 derniers jours pour la métrique sélectionnée. Par défaut, ce graphique affiche les 10 premiers éléments, mais vous pouvez réduire ce nombre à 3 éléments ou l'augmenter à 50 éléments.
- Un histogramme qui présente un graphique à barres verticales pour la date et la métrique sélectionnées. Si vous avez un grand nombre d'éléments à afficher dans ce graphique, vous devrez peut-être le faire défiler horizontalement.
- Un graphique d'analyse à bulles qui trace tous les éléments de la dimension. Ce graphique représente la première métrique sur l'axe des X et la deuxième sur l'axe des Y. La troisième métrique est représentée par la taille de la bulle.
- Une vue des métriques sous forme de grille qui contient chaque élément de la dimension répertoriés en lignes. Les colonnes représentent chaque métrique disponible, disposées dans des onglets de catégorie de métriques pour faciliter la navigation.

Afficher les métriques Amazon S3 Storage Lens à l'aide d'une exportation de données

Les métriques Amazon S3 Storage Lens sont générées quotidiennement dans des fichiers d'exportation au format CSV ou Apache Parquet et placées dans un compartiment S3 de votre compte. À partir de là, vous pouvez intégrer les statistiques exportées dans les outils d'analyse de

votre choix, tels qu'Amazon QuickSight et Amazon Athena, où vous pouvez analyser l'utilisation du stockage et les tendances en matière d'activité.

Rubriques

- Utilisation d'un AWS KMS key pour chiffrer vos exportations de métriques
- Qu'est-ce qu'un manifeste d'exportation S3 Storage Lens ?
- Comprendre le schéma d'exportation d'Amazon S3 Storage Lens

Utilisation d'un AWS KMS key pour chiffrer vos exportations de métriques

Pour accorder à Amazon S3 Storage Lens l'autorisation de chiffrer vos exportations de métriques à l'aide d'une clé gérée par le client, vous devez utiliser une politique de clé. Pour mettre à jour votre politique de clé et pouvoir utiliser une clé KMS permettant de chiffrer vos exportations de métriques S3 Storage Lens, procédez comme suit.

Pour accorder des autorisations S3 Storage Lens afin de chiffrer des données à l'aide de votre clé KMS

1. Connectez-vous au en AWS Management Console utilisant le propriétaire Compte AWS de la clé gérée par le client.
2. Ouvrez la AWS KMS console à l'adresse https://console.aws.amazon.com/kms.
3. Pour modifier le Région AWS, utilisez le sélecteur de région dans le coin supérieur droit de la page.
4. Dans le panneau de navigation de gauche, choisissez Clés gérées par le client.
5. Sous Clés gérées par le client, choisissez la clé que vous souhaitez utiliser pour chiffrer les exportations de métriques. AWS KMS keys sont spécifiques à une région et doivent se trouver dans la même région que le compartiment S3 de destination d'exportation des métriques.
6. Sous Politique de clé, choisissez Passer à l'écran de la politique.
7. Pour mettre à jour la politique de clé, choisissez Modifier.
8. Sous Modifier la politique de clé, ajoutez la politique de clé suivante à la politique de clé existante. Pour utiliser cette politique, remplacez *user input placeholders* par vos informations.

```
{
    "Sid": "Allow Amazon S3 Storage Lens use of the KMS key",
     "Effect": "Allow",
```

```
    "Principal": {
        "Service": "storage-lens.s3.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
       "StringEquals": {
           "aws:SourceArn": "arn:aws:s3:us-east-1:source-account-id:storage-
lens/your-dashboard-name",
           "aws:SourceAccount": "source-account-id"
        }
     }
}
```

9. Sélectionnez Enregistrer les modifications.

Pour en savoir plus sur la création de clés gérées par le client et l'utilisation des politiques de clés, consultez les rubriques suivantes dans le guide du développeur AWS Key Management Service :

- Prise en main
- Utilisation de politiques clés dans AWS KMS

Vous pouvez également utiliser l'opération d'API de politique AWS KMS PUT clé (PutKeyPolicy) pour copier la politique clé dans les clés gérées par le client que vous souhaitez utiliser pour chiffrer les exportations de métriques à l'aide de l'API REST et des SDK. AWS CLI

Qu'est-ce qu'un manifeste d'exportation S3 Storage Lens ?

Compte tenu de la grande quantité de données agrégées, l'exportation quotidienne de métriques S3 Storage Lens peut être divisée en plusieurs fichiers. Le fichier manifeste `manifest.json` décrit où se trouvent les fichiers d'exportation des métriques d'un jour donné. Chaque fois qu'une nouvelle exportation est livrée, elle est accompagnée d'un nouveau manifeste. Chaque manifeste contenu dans le fichier `manifest.json` fournit des métadonnées et d'autres informations de base sur l'exportation.

Les informations du manifeste comprennent les propriétés suivantes :

- `sourceAccountId` — ID de compte du propriétaire de la configuration.

- `configId` — Identifiant unique du tableau de bord.
- `destinationBucket` — Compartiment de destination Amazon Resource Name (ARN) dans lequel l'exportation des métriques est placée.
- `reportVersion` — Version de l'exportation.
- `reportDate` — Date du rapport.
- `reportFormat` — Format du rapport.
- `reportSchema` — Schéma du rapport.
- `reportFiles` — Liste réelle des fichiers de rapport d'exportation qui se trouvent dans le compartiment de destination.

Voici un exemple de manifeste dans un fichier `manifest.json` pour une exportation au format CSV.

```
{
   "sourceAccountId":"123456789012",
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"CSV",

 "reportSchema":"version_number,configuration_id,report_date,aws_account_number,aws_region,stor
   "reportFiles":[
        {
            "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-
configuration-id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
            "size":1603959,
            "md5Checksum":"2177e775870def72b8d84febe1ad3574"
        }
      ]
}
```

Voici un exemple de manifeste dans un fichier `manifest.json` pour une exportation au format Parquet.

```
{
   "sourceAccountId":"123456789012",
```

```
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"Parquet",
   "reportSchema":"message s3.storage.lens { required string version_number;
 required string configuration_id; required string report_date; required string
 aws_account_number; required string aws_region; required string storage_class;
 required string record_type; required string record_value; required string
 bucket_name; required string metric_name; required long metric_value; }",
   "reportFiles":[
      {
         "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-
id/V_1/reports/dt=2020-11-03/bd23de7c-b46a-4cf4-bcc5-b21aac5be0f5.par",
         "size":14714,
         "md5Checksum":"b5c741ee0251cd99b90b3e8eff50b944"
      }
}
```

Vous pouvez configurer l'exportation de vos métriques pour qu'elle soit générée dans le cadre de la configuration de votre tableau de bord dans la console Amazon S3 ou à l'aide de l'API REST Amazon S3 et des kits SDK. AWS CLI

Comprendre le schéma d'exportation d'Amazon S3 Storage Lens

Le tableau suivant présente le schéma de l'exportation des métriques S3 Storage Lens.

Nom d'attribut	Type de données	Nom de la colonne	Description
`VersionNumber`	Chaîne	`version_number`	Version des métriques de S3 Storage Lens en cours d'utilisation.
`ConfigurationId`	Chaîne	`configuration_id`	`configuration_id` de votre configuration S3 Storage Lens.
`ReportDate`	Chaîne	`report_date`	Date à laquelle les métriques ont été suivies.

Nom d'attribut	Type de données	Nom de la colonne	Description
`AwsAccountNumber`	Chaîne	`aws_account_number`	Ton Compte AWS numéro.
`AwsRegion`	Chaîne	`aws_region`	Les indicateurs Région AWS pour lesquels les mesures sont suivies.
`StorageClass`	Chaîne	`storage_class`	Classe de stockage du compartiment en question.
`RecordType`	ENUM	`record_type`	Type d'artefact indiqué (ACCOUNT, BUCKET ou PREFIX).
`RecordValue`	Chaîne	`record_value`	La valeur de l'artefact `RecordType` . **Note** La valeur `record_value` est codée dans l'URL.
`BucketName`	Chaîne	`bucket_name`	Nom du compartiment indiqué.
`MetricName`	Chaîne	`metric_name`	Nom de la métrique indiquée.
`MetricValue`	Long	`metric_value`	Valeur de la métrique indiquée.

Exemple d'exportation de métriques S3 Storage Lens

Voici un exemple d'exportation de métriques S3 Storage Lens basée sur ce schéma.

> **Note**
>
> Vous pouvez identifier les métriques pour les groupes Storage Lens en recherchant les valeurs `STORAGE_LENS_GROUP_BUCKET` ou `STORAGE_LENS_GROUP_ACCOUNT` ou dans la colonne `record_type`. La colonne `record_value` affichera l'Amazon Resource Name (ARN) du groupe Storage Lens, par exemple `arn:aws:s3:us-east-1:123456789012:storage-lens-group/slg-1`.

version_r	configuration_id	report_date	aws_account_number	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			StorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedStorageBytes	20000
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjectCount	20
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedStorageBytes	2478828742
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjectCount	1598961
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			DeleteMarkerObjectCount	1500
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionStorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	StorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	DeleteMarkerObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadObjectCount	0

Voici un exemple d'exportation de métriques S3 Storage Lens avec des données de groupes Storage Lens.

version_numbe	configuration_id	report_date	aws_account_num	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		StorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUObjectCountOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	StorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUObjectCountOlderThan7Days	0

Surveillance des métriques S3 Storage Lens dans CloudWatch

Vous pouvez publier les métriques S3 Storage Lens sur Amazon CloudWatch pour créer une vue unifiée de votre état opérationnel dans les tableaux de bord CloudWatch. Vous pouvez également utiliser les fonctionnalités CloudWatch, comme les alarmes et les actions déclenchées, les mathématiques appliquées aux métriques et la détection d'anomalies pour surveiller les métriques S3 Storage Lens et agir dessus. En outre, les opérations d'API CloudWatch permettent aux applications, y compris à celles des fournisseurs tiers, d'accéder à vos métriques S3 Storage Lens. Pour de plus amples informations sur les fonctions de CloudWatch, consultez le guide de l'utilisateur Amazon CloudWatch.

Vous pouvez activer l'option de publication sur CloudWatch pour les configurations de tableau de bord nouvelles ou existantes à l'aide de la console Amazon S3, de l'API REST Amazon S3, d'AWS CLI et des kits AWS SDK. Les tableaux de bord mis à niveau vers les métriques et recommandations avancées S3 Storage Lens peuvent utiliser l'option CloudWatch publishing (publication sur CloudWatch). Pour obtenir la tarification des métriques et recommandations avancées S3 Storage Lens, consultez Tarification Amazon S3. Aucun frais supplémentaire de publication de métriques CloudWatch ne s'applique. Toutefois, d'autres frais CloudWatch comme les frais de tableaux de bord, d'alarmes et d'appels d'API s'appliquent. Pour de plus amples informations, consultez Tarification Amazon CloudWatch.

Les métriques S3 Storage Lens sont publiées sur CloudWatch dans le compte propriétaire de la configuration S3 Storage Lens. Une fois que vous avez activé l'option de publication sur CloudWatch

dans les métriques et recommandations avancées, vous pouvez accéder aux métriques au niveau de l'organisation, du compte et du compartiment dans CloudWatch. Les métriques au niveau du préfixe ne sont pas disponibles dans CloudWatch.

> **Note**
>
> Les métriques S3 Storage Lens sont des métriques quotidiennes et sont publiées sur CloudWatch une fois par jour. Lorsque vous interrogez des métriques S3 Storage Lens dans CloudWatch, la période de la requête doit être d'un jour (86 400 secondes). Une fois que vos métriques quotidiennes S3 Storage Lens s'affichent dans votre tableau de bord S3 Storage Lens de la console Amazon S3, plusieurs heures peuvent être nécessaires pour que ces mêmes métriques s'affichent dans CloudWatch. Lorsque vous activez l'option CloudWatch publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens pour la première fois, la publication de vos métriques sur CloudWatch peut prendre jusqu'à 24 heures.

Après avoir activé l'option CloudWatch publishing (publication sur CloudWatch), vous pouvez utiliser les fonctions CloudWatch suivantes pour surveiller et analyser les données S3 Storage Lens :

- Tableaux de bord : utilisez les tableaux de bord CloudWatch pour créer des tableaux de bord S3 Storage Lens personnalisés. Partagez votre tableau de bord CloudWatch avec des personnes qui n'ont pas d'accès direct à votre Compte AWS, entre des équipes, avec des parties prenantes et avec des personnes extérieures à vos organisations.
- Alarmes et actions déclenchées : configurez les alarmes qui surveillent les métriques et agissent lorsqu'un seuil est dépassé. Par exemple, vous pouvez configurer une alarme qui envoie une notification Amazon SNS lorsque la métrique Incomplete Multipart Upload Bytes (Octets de chargements partitionnés non terminés) dépasse 1 Go pendant trois jours consécutifs.
- Détection des anomalies : activez la détection des anomalies pour analyser les métriques continuellement, déterminer l'activité normale et les anomalies de surface. Vous pouvez créer une alarme de détection d'anomalie basées sur la valeur attendue d'une métrique. Par exemple, vous pouvez surveiller les anomalies pour la métrique Object Lock Enabled Bytes (Octets activés par le verrouillage d'objets) afin de détecter la suppression non autorisée des paramètres de verrouillage d'objets.
- Mathématiques appliquées aux métriques : vous pouvez également utiliser les mathématiques appliquées aux métriques pour interroger plusieurs métriques S3 Storage Lens et utiliser des expressions mathématiques pour créer des séries temporelles basées sur ces métriques. Par

exemple, vous pouvez créer une nouvelle métrique pour obtenir la taille moyenne d'objet en divisant `StorageBytes` par `ObjectCount`.

Pour plus d'informations sur l'option CloudWatch publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens, consultez les rubriques ci-dessous.

Rubriques

- Métriques et dimensions S3 Storage Lens
- Activation de la publication sur CloudWatch pour S3 Storage Lens
- Utilisation des métriques S3 Storage Lens dans CloudWatch

Métriques et dimensions S3 Storage Lens

Pour envoyer des métriques S3 Storage Lens à CloudWatch, vous devez activer l'option CloudWatch publishing (publication sur CloudWatch) dans les métriques et recommandations avancées S3 Storage Lens. Une fois les métriques avancées activées, vous pouvez utiliser les tableaux de bord CloudWatch pour surveiller les métriques S3 Storage Lens en même temps que d'autres métriques d'application et créer une vue unifiée de votre état opérationnel. Vous pouvez utiliser des dimensions pour filtrer vos métriques S3 Storage Lens dans CloudWatch par organisation, compte, compartiment, classe de stockage, Région et ID de configuration de métriques.

Les métriques S3 Storage Lens sont publiées sur CloudWatch dans le compte propriétaire de la configuration S3 Storage Lens. Une fois que vous avez activé l'option de publication sur CloudWatch dans les métriques et recommandations avancées, vous pouvez accéder aux métriques au niveau de l'organisation, du compte et du compartiment dans CloudWatch. Les métriques au niveau du préfixe ne sont pas disponibles dans CloudWatch.

> **Note**
>
> Les métriques S3 Storage Lens sont des métriques quotidiennes et sont publiées sur CloudWatch une fois par jour. Lorsque vous interrogez des métriques S3 Storage Lens dans CloudWatch, la période de la requête doit être d'un jour (86 400 secondes). Une fois que vos métriques quotidiennes S3 Storage Lens s'affichent dans votre tableau de bord S3 Storage Lens de la console Amazon S3, plusieurs heures peuvent être nécessaires pour que ces mêmes métriques s'affichent dans CloudWatch. Lorsque vous activez l'option CloudWatch

publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens pour la première fois, la publication de vos métriques sur CloudWatch peut prendre jusqu'à 24 heures.

Pour plus d'informations sur les métriques et dimensions S3 Storage Lens dans CloudWatch, consultez les rubriques ci-dessous.

Rubriques

- Métriques
- Dimensions

Métriques

Les métriques S3 Storage Lens sont disponibles sous forme de métriques dans CloudWatch. Les métriques S3 Storage Lens sont publiées sur l'espace de noms `AWS/S3/Storage-Lens`. Cet espace de noms est réservé aux métriques S3 Storage Lens. Les métriques de compartiment, de demande et de réplication Amazon S3 sont publiées sur l'espace de noms `AWS/S3`.

Les métriques S3 Storage Lens sont publiées sur CloudWatch dans le compte propriétaire de la configuration S3 Storage Lens. Une fois que vous avez activé l'option de publication sur CloudWatch dans les métriques et recommandations avancées, vous pouvez accéder aux métriques au niveau de l'organisation, du compte et du compartiment dans CloudWatch. Les métriques au niveau du préfixe ne sont pas disponibles dans CloudWatch.

Dans S3 Storage Lens, les métriques sont agrégées et stockées uniquement dans la Région d'origine désignée. Les métriques S3 Storage Lens sont également publiées sur CloudWatch dans la Région d'origine que vous spécifiez dans la configuration S3 Storage Lens.

Pour obtenir la liste complète des métriques S3 Storage Lens, y compris la liste des métriques disponibles dans CloudWatch, consultez le Glossaire des métriques Amazon S3 Storage Lens.

> **Note**
>
> La statistique valide pour les métriques S3 Storage Lens dans CloudWatch est Average (Moyenne). Pour plus d'informations sur les statistiques dans CloudWatch, consultez Définitions des statistiques CloudWatch dans le guide de l'utilisateur Amazon CloudWatch.

Granularité des métriques S3 Storage Lens dans CloudWatch

S3 Storage Lens offre des métriques selon une granularité au niveau de l'organisation, du compte, du compartiment et des préfixes. S3 Storage Lens publie les métriques S3 Storage Lens au niveau de l'organisation, du compte et du compartiment sur CloudWatch. Les métriques S3 Storage Lens au niveau du préfixe ne sont pas disponibles dans CloudWatch.

Pour plus d'informations sur la granularité des métriques S3 Storage Lens disponibles dans CloudWatch, consultez la liste suivante :

- Organisation : métriques agrégées sur les comptes membres de votre organisation. S3 Storage Lens publie des métriques pour les comptes membres sur CloudWatch dans le compte de gestion.
  - Organisation et compte : métriques pour les comptes membres de votre organisation.
  - Organisation et compartiment : métriques pour les compartiments Amazon S3 dans les comptes membres de votre organisation.
- Compte (hors de l'organisation) : mesures agrégées dans les compartiments de votre compte.
- Compartiment (hors de l'organisation) : métriques pour un compartiment spécifique. Dans CloudWatch, S3 Storage Lens publie ces mesures sur le Compte AWS qui a créé la configuration S3 Storage Lens. S3 Storage Lens publie ces métriques uniquement pour les configurations non organisationnelles.

Dimensions

Quand S3 Storage Lens envoie des données à CloudWatch, les dimensions sont attachées à chaque métrique. Les dimensions sont des catégories qui décrivent les caractéristiques des métriques. Vous pouvez utiliser des dimensions pour filtrer les résultats renvoyés par CloudWatch.

Par exemple, toutes les métriques S3 Storage Lens dans CloudWatch sont rangées dans la dimension `configuration_id`. Vous pouvez utiliser cette dimension pour différencier les métriques associées à une configuration spécifique de S3 Storage Lens. La dimension `organization_id` identifie les métriques au niveau de l'organisation. Pour plus d'informations sur les dimensions dans CloudWatch, consultez la rubrique Dimensions dans le guide de l'utilisateur CloudWatch.

Différentes dimensions sont disponibles pour les métriques S3 Storage Lens selon la granularité des métriques. Par exemple, vous pouvez utiliser la dimension `organization_id` pour filtrer les métriques au niveau de l'organisation par l'ID AWS Organizations. Toutefois, vous ne pouvez pas

utiliser cette dimension pour les métriques au niveau du compartiment ni du compte. Pour de plus amples informations, veuillez consulter Filtrage des métriques à l'aide de dimensions.

Pour savoir quelles dimensions sont disponibles pour votre configuration S3 Storage Lens, consultez le tableau suivant.

Dimension	Description	Compartiment	Organisation et compartiment	Organisation et compte	Organisation et compte
`configuration_id`	Nom du tableau de bord pour la configuration S3 Storage Lens indiqué dans les métriques				
`metrics_version`	Version des métriques S3 Storage Lens. La valeur fixe de la version des métriques est égale à `1.0`.				
`organization_id`	ID AWS Organizations des métriques				
`aws_account_number`	Le Compte AWS associé aux métriques				
`aws_region`	La Région AWS pour les métriques				
`bucket_name`	Nom du compartiment S3 indiqué dans les métriques				
`storage_class`	Classe de stockage pour le compartiment indiqué dans les métriques				
`record_type`	Granularité des métriques : ORGANISATION, COMPTE, COMPARTIMENT	COMPARTIMENT	COMPTE	COMPARTIMENT	ORGANISATION

Activation de la publication sur CloudWatch pour S3 Storage Lens

Vous pouvez publier les métriques S3 Storage Lens sur Amazon CloudWatch pour créer une vue unifiée de votre état opérationnel dans les tableaux de bord CloudWatch. Vous pouvez également utiliser les fonctionnalités CloudWatch, comme les alarmes et les actions déclenchées, les mathématiques appliquées aux métriques et la détection d'anomalies pour surveiller les métriques S3 Storage Lens et agir dessus. En outre, les opérations d'API CloudWatch permettent aux applications, y compris à celles des fournisseurs tiers, d'accéder à vos métriques S3 Storage Lens. Pour de plus amples informations sur les fonctions de CloudWatch, consultez le guide de l'utilisateur Amazon CloudWatch.

Les métriques S3 Storage Lens sont publiées sur CloudWatch dans le compte propriétaire de la configuration S3 Storage Lens. Une fois que vous avez activé l'option de publication sur CloudWatch dans les métriques et recommandations avancées, vous pouvez accéder aux métriques au niveau de l'organisation, du compte et du compartiment dans CloudWatch. Les métriques au niveau du préfixe ne sont pas disponibles dans CloudWatch.

Vous pouvez activer la prise en charge CloudWatch pour des configurations de tableau de bord nouvelles ou existantes à l'aide de la console S3, des API REST Amazon S3, d'AWS CLI et des kits AWS SDK. L'option de publication sur CloudWatch est disponible pour les tableaux de bord mis à niveau vers les métriques et recommandations avancées S3 Storage Lens. Pour obtenir la tarification des métriques et recommandations avancées S3 Storage Lens, consultez Tarification Amazon S3. Aucun frais supplémentaire de publication de métriques CloudWatch ne s'applique. Toutefois, d'autres frais CloudWatch comme les frais de tableaux de bord, d'alarmes et d'appels d'API s'appliquent.

Pour activer l'option CloudWatch publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens, consultez les rubriques ci-dessous.

> **Note**
>
> Les métriques S3 Storage Lens sont des métriques quotidiennes et sont publiées sur CloudWatch une fois par jour. Lorsque vous interrogez des métriques S3 Storage Lens dans CloudWatch, la période de la requête doit être d'un jour (86 400 secondes). Une fois que vos métriques quotidiennes S3 Storage Lens s'affichent dans votre tableau de bord S3 Storage Lens de la console Amazon S3, plusieurs heures peuvent être nécessaires pour que ces mêmes métriques s'affichent dans CloudWatch. Lorsque vous activez l'option CloudWatch publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens pour la première fois, la publication de vos métriques sur CloudWatch peut prendre jusqu'à 24 heures.

Actuellement, les métriques S3 Storage Lens ne peuvent pas être utilisées via les flux CloudWatch.

Utilisation de la console S3

Lorsque vous mettez à jour un tableau de bord S3 Storage Lens, vous ne pouvez pas modifier le nom du tableau de bord ni la région d'origine. Vous ne pouvez pas non plus modifier la portée du tableau de bord par défaut, qui est délimitée à l'ensemble du stockage de votre compte.

Pour mettre à jour un tableau de bord S3 Storage Lens afin d'activer la publication sur CloudWatch

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez S3 Storage Lens, Dashboards (Tableaux de bord).
3. Choisissez le tableau de bord que vous souhaitez modifier, puis choisissez Edit (Modifier).
4. Sous Metrics selection (Sélection des métriques), choisissez Advanced metrics and recommendations (Métriques et recommandations avancées).

   Des métriques et recommandations avancées sont disponibles moyennant des frais supplémentaires. Les métriques et recommandations avancées incluent une période de 15 mois pour les requêtes de données, les métriques d'utilisation agrégées au niveau du préfixe, les métriques d'activité agrégées par compartiment, l'option de publication sur CloudWatch et les recommandations contextuelles qui vous aident à optimiser les coûts de stockage et à appliquer les bonnes pratiques de protection des données. Pour de plus amples informations, veuillez consulter Tarification Amazon S3.
5. Sous Select Advanced metrics and recommendations features (Sélectionner les métriques et les recommandations avancées), sélectionnez CloudWatch publishing (Publication sur CloudWatch).

   Important

   Si votre configuration autorise l'agrégation de préfixes pour les métriques d'utilisation, les métriques au niveau du préfixe ne seront pas publiées sur CloudWatch. Seules les métriques S3 Storage Lens au niveau du compartiment, du compte et de l'organisation sont publiées sur CloudWatch.
6. Sélectionnez Save Changes (Enregistrer les modifications).

Création d'un tableau de bord S3 Storage Lens qui permet la prise en charge de CloudWatch

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Choisissez Create dashboard (Créer un tableau de bord).
4. Sous General (Général), définissez les options de configuration suivantes :

   a. Dans Dashboard name (Nom du tableau de bord), saisissez le nom de votre tableau de bord.

      Les noms de tableau de bord doivent contenir moins de 65 caractères et ne doivent pas contenir de caractères spéciaux ou d'espaces. Vous ne pouvez pas modifier le nom d'un tableau de bord une fois que vous l'avez créé.

   b. Sélectionnez la Home Region (Région d'origine) de votre tableau de bord.

      Les métriques de toutes les Régions incluses dans la portée de ce tableau de bord sont stockées de manière centralisée dans la Région d'origine désignée. Dans CloudWatch, les métriques S3 Storage Lens sont également disponibles dans la Région d'origine. Vous ne pouvez pas modifier la région d'origine une fois que vous avez créé votre tableau de bord.
5. (Facultatif) Pour ajouter des étiquettes, choisissez Add tag (Ajouter une balise) et saisissez la Key (Clé) d'étiquette et une Value (Valeur).

   > **Note**
   >
   > Vous pouvez ajouter jusqu'à 50 balises à votre configuration de tableau de bord.
6. Définissez la portée de votre configuration :

   a. Si vous créez une configuration au niveau de l'organisation, choisissez les comptes à inclure dans la configuration : Include all accounts in your configuration (Inclure tous les comptes dans votre configuration) ou Limit the scope to your signed-in account (Limiter la portée au compte connecté).

> **Note**
>
> Lorsque vous créez une configuration au niveau de l'organisation qui inclut tous les comptes, vous pouvez inclure ou exclure uniquement des régions, et non pas des compartiments.

b. Choisissez les régions et les compartiments que vous souhaitez que S3 Storage Lens inclue de la configuration du tableau de bord en procédant comme suit :

- Pour inclure toutes les Régions, choisissez Include Regions and buckets (Inclure les Régions et les compartiments).
- Pour inclure des Régions spécifiques, effacez Include all Regions (Inclure toutes les Régions). Sous Choose Regions to include (Choisir les Régions à inclure), choisissez les Régions que vous souhaitez que S3 Storage Lens inclue dans le tableau de bord.
- Pour inclure des compartiments spécifiques, effacez Include all buckets (Inclure tous les compartiments). Sous Choose buckets to include (Choisir les compartiments à inclure), choisissez les compartiments que vous souhaitez que S3 Storage Lens inclue dans le tableau de bord.

> **Note**
>
> Vous pouvez choisir jusqu'à 50 compartiments.

7. Pour Metrics selection (Sélection des métriques), choisissez Advanced metrics and recommendations (Métriques et recommandations avancées).

   Pour plus d'informations sur la tarification des métriques et recommandations avancées, consultez Tarification Amazon S3.

8. Sous Advanced metrics and recommendations features (Fonctionnalités de métriques et recommandations avancées), sélectionnez les options que vous voulez activer :

   - Advanced metrics (Métriques avancées)
   - CloudWatch publishing (Publication sur CloudWatch)

> **Important**
>
> Si vous activez l'agrégation de préfixes pour votre configuration S3 Storage Lens, les métriques au niveau du préfixe ne seront pas publiées sur CloudWatch. Seules les métriques S3 Storage Lens au niveau du compartiment, du compte et de l'organisation sont publiées sur CloudWatch.

- Prefix aggregation (Agrégation de préfixes)

> **Note**
>
> Pour plus d'informations sur les fonctionnalités des métriques et recommandations avancées, consultez Sélection des métriques.

9. Si vous avez activé Advanced metrics (Métriques avancées), sélectionnez les Advanced metrics categories (Catégories de métriques avancées) que vous souhaitez afficher dans votre tableau de bord S3 Storage Lens :

   - Métriques d'activité
   - Detailed status code metrics (Métriques détaillées sur le code de statut)
   - Advanced cost optimization metrics (Métriques avancées sur l'optimisation des coûts)
   - Advanced data protection metrics (Métriques avancées sur la protection des données)

   Pour plus d'informations sur les catégories de métriques, consultez Catégories de métriques. Pour obtenir une liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.

10. (Facultatif) Configurez l'exportation de vos métriques.

    Pour plus d'informations sur la configuration d'une exportation de métriques, consultez l'étape Créer un tableau de bord Amazon S3 Storage Lens.

11. Choisissez Create dashboard (Créer un tableau de bord).

Utilisation de AWS CLI

L'exemple AWS CLI suivant permet d'activer l'option de publication sur CloudWatch à l'aide d'une configuration S3 Storage Lens au niveau de l'organisation des métriques et recommandations avancées. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
aws s3control put-storage-lens-configuration --account-id=555555555555 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json

config.json
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
```

```
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true, //Mark this as false if you want only free metrics.
          "SelectionCriteria":{
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
 configurations.
      "arn:aws:s3:::source_bucket1"
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::destination-bucket-name", // The destination bucket for your
 metrics export must be in the same Region as your S3 Storage Lens configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true //Mark this as false if you want to export only free metrics.
    }
  }
}
```

Utilisation du kit AWS SDK pour Java

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
```

```
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            Include include = new Include()
                    .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
                    .withRegions(Arrays.asList("us-west-2"));

            StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                    .withSSES3(new SSES3());
            S3BucketDestination s3BucketDestination = new S3BucketDestination()
                    .withAccountId(exportAccountId)
                    .withArn(exportBucketArn)
                    .withEncryption(exportEncryption)
                    .withFormat(exportFormat)
                    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
```

```
                .withPrefix("Prefix");
            CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                .withIsEnabled(true);
            StorageLensDataExport dataExport = new StorageLensDataExport()
                .withCloudWatchMetrics(cloudWatchMetrics)
                .withS3BucketDestination(s3BucketDestination);

            StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                .withArn(awsOrgARN);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withInclude(include)
                .withDataExport(dataExport)
                .withAwsOrg(awsOrg)
                .withIsEnabled(true);

            List<StorageLensTag> tags = Arrays.asList(
                new StorageLensTag().withKey("key-1").withValue("value-1"),
                new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();

            s3ControlClient.putStorageLensConfiguration(new
PutStorageLensConfigurationRequest()
                .withAccountId(sourceAccountId)
                .withConfigId(configurationId)
                .withStorageLensConfiguration(configuration)
                .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
```

```
    }
}
```

Utilisation de l'API REST

Pour activer l'option de publication sur CloudWatch à l'aide de l'API REST Amazon S3, vous pouvez utiliser PutStorageLensConfiguration.

Étapes suivantes

Une fois que vous avez activé l'option CloudWatch publishing (publication sur CloudWatch), vous pouvez accéder à vos métriques S3 Storage Lens dans CloudWatch. Vous pouvez également exploiter les fonctions CloudWatch pour surveiller et analyser vos données S3 Storage Lens dans CloudWatch. Pour plus d'informations, consultez les rubriques suivantes :

- Métriques et dimensions S3 Storage Lens
- Utilisation des métriques S3 Storage Lens dans CloudWatch

Utilisation des métriques S3 Storage Lens dans CloudWatch

Vous pouvez publier les métriques S3 Storage Lens sur Amazon CloudWatch pour créer une vue unifiée de votre état opérationnel dans les tableaux de bord CloudWatch. Vous pouvez également utiliser les fonctionnalités CloudWatch, comme les alarmes et les actions déclenchées, les mathématiques appliquées aux métriques et la détection d'anomalies pour surveiller les métriques S3 Storage Lens et agir dessus. En outre, les opérations d'API CloudWatch permettent aux applications, y compris à celles des fournisseurs tiers, d'accéder à vos métriques S3 Storage Lens. Pour de plus amples informations sur les fonctions de CloudWatch, consultez le guide de l'utilisateur Amazon CloudWatch.

Vous pouvez activer l'option de publication sur CloudWatch pour les configurations de tableau de bord nouvelles ou existantes à l'aide de la console Amazon S3, des API REST Amazon S3, d'AWS CLI et des kits AWS SDK. L'option de publication sur CloudWatch est disponible pour les tableaux de bord mis à niveau vers les métriques et recommandations avancées S3 Storage Lens. Pour obtenir la tarification des métriques et recommandations avancées S3 Storage Lens, consultez Tarification Amazon S3. Aucun frais supplémentaire de publication de métriques CloudWatch ne s'applique. Toutefois, d'autres frais CloudWatch comme les frais de tableaux de bord, d'alarmes et d'appels d'API s'appliquent. Pour de plus amples informations, consultez Tarification Amazon CloudWatch.

Les métriques S3 Storage Lens sont publiées sur CloudWatch dans le compte propriétaire de la configuration S3 Storage Lens. Une fois que vous avez activé l'option de publication sur CloudWatch dans les métriques et recommandations avancées, vous pouvez accéder aux métriques au niveau de l'organisation, du compte et du compartiment dans CloudWatch. Les métriques au niveau du préfixe ne sont pas disponibles dans CloudWatch.

> **Note**
>
> Les métriques S3 Storage Lens sont des métriques quotidiennes et sont publiées sur CloudWatch une fois par jour. Lorsque vous interrogez des métriques S3 Storage Lens dans CloudWatch, la période de la requête doit être d'un jour (86 400 secondes). Une fois que vos métriques quotidiennes S3 Storage Lens s'affichent dans votre tableau de bord S3 Storage Lens de la console Amazon S3, plusieurs heures peuvent être nécessaires pour que ces mêmes métriques s'affichent dans CloudWatch. Lorsque vous activez l'option CloudWatch publishing (publication sur CloudWatch) pour les métriques S3 Storage Lens pour la première fois, la publication de vos métriques sur CloudWatch peut prendre jusqu'à 24 heures. Actuellement, les métriques S3 Storage Lens ne peuvent pas être utilisées via les flux CloudWatch.

Pour plus d'informations sur l'utilisation des métriques S3 Storage Lens dans CloudWatch, consultez les rubriques suivantes.

Rubriques

- Utilisation des tableaux de bord CloudWatch
- Définition d'alarmes, déclenchement d'actions et utilisation de la détection d'anomalies
- Filtrage des métriques à l'aide de dimensions
- Calcul de nouvelles métriques avec des mathématiques appliquées aux métriques
- Utilisation d'expressions de recherche dans les graphiques

Utilisation des tableaux de bord CloudWatch

Vous pouvez utiliser les tableaux de bord CloudWatch pour surveiller les métriques S3 Storage Lens en même temps que d'autres métriques d'application et créer une vue unifiée de l'état opérationnel. Les tableaux de bord sont des pages d'accueil personnalisables dans la console CloudWatch, qui permettent de contrôler vos ressources dans une même vue.

CloudWatch dispose d'un contrôle étendu des autorisations qui ne prend pas en charge la limitation de l'accès à un ensemble spécifique de métriques ou de dimensions. Les utilisateurs de votre compte ou de votre organisation qui ont accès à CloudWatch auront accès aux métriques de toutes les configurations S3 Storage Lens pour lesquelles l'option de prise en charge de CloudWatch est activée. Vous ne pouvez pas gérer les autorisations pour des tableaux de bord spécifiques comme dans S3 Storage Lens. Pour plus d'informations sur les autorisations CloudWatch, consultez la section Gestion des autorisations d'accès à vos ressources CloudWatch dans le guide de l'utilisateur Amazon CloudWatch.

Pour plus d'informations sur l'utilisation des tableaux de bord CloudWatch et sur la configuration des autorisations, consultez les sections Utilisation des tableaux de bord Amazon CloudWatch et Partage de tableaux de bord CloudWatch dans le guide de l'utilisateur Amazon CloudWatch.

Définition d'alarmes, déclenchement d'actions et utilisation de la détection d'anomalies

Vous pouvez configurer des alarmes CloudWatch qui surveillent les métriques S3 Storage Lens dans CloudWatch et agissent lorsqu'un seuil est dépassé. Par exemple, vous pouvez configurer une alarme qui envoie une notification Amazon SNS lorsque la métrique Incomplete Multipart Upload Bytes (Octets de chargements partitionnés non terminés) dépasse 1 Go pendant trois jours consécutifs.

Vous pouvez également activer la détection des anomalies pour analyser continuellement les métriques S3 Storage Lens, déterminer les références normales et les anomalies de surface. Vous pouvez créer une alarme de détection d'anomalie basée sur la valeur attendue d'une métrique. Par exemple, vous pouvez surveiller les anomalies pour la métrique Object Lock Enabled Bytes (Octets activés par le verrouillage d'objets) afin de détecter la suppression non autorisée des paramètres de verrouillage d'objets.

Pour plus d'informations et d'exemples, consultez les sections Utilisation des alarmes Amazon CloudWatch et Création d'une alarme à partir d'une métrique d'un graphique dans leguide de l'utilisateur Amazon CloudWatch.

Filtrage des métriques à l'aide de dimensions

Vous pouvez utiliser des dimensions pour filtrer les métriques S3 Storage Lens dans la console CloudWatch. Par exemple, vous pouvez filtrer par `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` et plus encore.

S3 Storage Lens prend en charge plusieurs configurations de tableau de bord par compte. Cela signifie que différentes configurations peuvent inclure le même compartiment. Lorsque ces

métriques sont publiées dans CloudWatch, le compartiment comporte des métriques en double dans CloudWatch. Pour afficher uniquement les métriques d'une configuration spécifique de S3 Storage Lens dans CloudWatch, vous pouvez utiliser la dimension `configuration_id`. Lorsque vous filtrez par `configuration_id`, vous ne voyez que les métriques associées à la configuration que vous identifiez.

Pour plus d'informations sur le filtrage par ID de configuration, consultez Recherche de métriques disponibles dans le Guide de l'utilisateur Amazon CloudWatch.

Calcul de nouvelles métriques avec des mathématiques appliquées aux métriques

Vous pouvez utiliser les mathématiques appliquées aux métriques pour interroger plusieurs métriques S3 Storage Lens et utiliser des expressions mathématiques pour créer des séries temporelles basées sur ces métriques. Par exemple, vous pouvez créer une nouvelle métrique pour les objets non chiffrés en soustrayant les objets chiffrés du nombre d'objets. Vous pouvez également créer une métrique pour obtenir la taille d'objet moyenne en divisant `StorageBytes` par `ObjectCount` ou le nombre d'octets consultés en un jour en divisant `BytesDownloaded` par `StorageBytes`.

Pour plus d'informations, consultez la section Utilisation des métriques Amazon CloudWatch dans le guide de l'utilisateur Amazon CloudWatch.

Utilisation d'expressions de recherche dans les graphiques

Avec les métriques S3 Storage Lens, vous pouvez créer une expression de recherche. Par exemple, vous pouvez créer une expression de recherche pour toutes les métriques nommées IncompleteMultipartUploadStorageBytes et ajouter `SUM` à cette expression. Avec cette expression de recherche, vous pouvez voir le nombre total d'octets de chargements partitionnés non terminés dans toutes les dimensions de votre stockage dans une seule métrique.

Cet exemple montre la syntaxe que vous utiliseriez pour créer une expression de recherche pour toutes les métriques nommées IncompleteMultipartUploadStorageBytes.

```
SUM(SEARCH('{AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}
 MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Pour plus d'informations sur cette syntaxe, consultez la section Syntaxe d'une expression de recherche CloudWatch dans le guide de l'utilisateur Amazon CloudWatch. Pour créer un

graphique CloudWatch avec une expression de recherche, consultez la section Création d'un graphique CloudWatch avec une expression de recherche dans le guide de l'utilisateur Amazon CloudWatch.

Cas d'utilisation des métriques Amazon S3 Storage Lens

Vous pouvez utiliser votre tableau de bord Amazon S3 Storage Lens pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations. Les métriques S3 Storage Lens sont organisées en catégories qui correspondent aux principaux cas d'utilisation. Vous pouvez utiliser ces métriques pour effectuer les opérations suivantes :

- Identifier les opportunités d'optimisation des coûts
- Appliquer les bonnes pratiques de protection des données
- Appliquer les bonnes pratiques de gestion des accès
- Améliorer les performances des charges de travail d'application

Par exemple, les métriques sur l'optimisation des coûts vous permettent d'identifier les opportunités pour réduire vos coûts de stockage Amazon S3. Vous pouvez identifier les compartiments contenant des chargements partitionnés datant de plus de 7 jours ou des compartiments qui accumulent d'anciennes versions.

De même, vous pouvez utiliser des métriques sur la protection des données pour identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données au sein de votre organisation. Par exemple, vous pouvez identifier les compartiments qui n'utilisent pas de clés AWS Key Management Service (SSE-KMS) pour le chiffrement par défaut ou pour lesquels la gestion des versions S3 n'est pas activée.

Grâce aux métriques de gestion des accès S3 Storage Lens, vous pouvez identifier les paramètres des compartiments pour la propriété de l'objet S3 afin de migrer les autorisations de liste de contrôle d'accès (ACL) vers les politiques des compartiments et de désactiver les listes ACL.

Si vous avez activé S3 Storage Lens advanced metrics (Métriques avancées S3 Storage Lens), vous pouvez utiliser des métriques de codes de statut détaillés pour obtenir les nombres de demandes ayant réussi et échoué que vous pouvez utiliser pour résoudre les problèmes d'accès et de performances.

Grâce aux métriques avancées, vous pouvez également accéder à des métriques supplémentaires sur l'optimisation des coûts et la protection des données que vous pouvez utiliser pour identifier les

opportunités de réduire encore vos coûts globaux de stockage S3 et de mieux vous aligner sur les bonnes pratiques pour protéger vos données. Par exemple, les métriques avancées sur l'optimisation des coûts incluent les nombres de règles de cycle de vie que vous pouvez utiliser pour identifier les compartiments qui ne disposent pas de règles de cycle de vie pour faire expirer les chargements partitionnés non terminés datant de plus de 7 jours. Les métriques avancées sur la protection des données incluent les nombres de règles de réplication.

Pour plus d'informations sur les catégories de métriques, consultez Catégories de métriques. Pour obtenir la liste complète des métriques S3 Storage Lens, consultez Glossaire des métriques Amazon S3 Storage Lens.

Rubriques

Utiliser Amazon S3 Storage Lens pour optimiser vos coûts de stockage

Vous pouvez utiliser les métriques sur l'optimisation des coûts S3 Storage Lens pour réduire le coût global de votre stockage S3. Les métriques sur l'optimisation des coûts peuvent vous aider à confirmer que vous avez configuré Amazon S3 de manière rentable et conformément aux bonnes pratiques. Par exemple, vous pouvez identifier les opportunités d'optimisation des coûts suivantes :

- Compartiments contenant des chargements partitionnés non terminés datant de plus de 7 jours
- Compartiments qui accumulent de nombreuses versions anciennes
- Compartiments dépourvus de règles de cycle de vie pour abandonner les chargements partitionnés non terminés
- Compartiments dépourvus de règles de cycle de vie pour faire expirer des objets de version ancienne
- Compartiments dépourvus de règles de cycle de vie pour transférer des objets vers une autre classe de stockage

Vous pouvez ensuite utiliser ces données pour ajouter des règles de cycle de vie supplémentaires à vos compartiments.

Les exemples suivants montrent comment utiliser les métriques sur l'optimisation des coûts dans votre tableau de bord S3 Storage Lens pour optimiser vos coûts de stockage.

Rubriques

Identifier vos compartiments S3 les plus importants

Des frais de stockage d'objets dans les compartiments S3 vous sont facturés. Le tarif qui vous est facturé dépend de la taille de vos objets, de leur durée de stockage et de leurs classes de stockage. Avec S3 Storage Lens, vous obtenez une vue centralisée de tous les compartiments figurant dans votre compte. Pour afficher tous les compartiments de tous les comptes de votre organisation, vous pouvez configurer un tableau de bord S3 Storage Lens de niveau AWS Organizations. À partir de cette vue de tableau de bord, vous pouvez identifier vos compartiments les plus importants.

Étape 1 : identifier vos plus grands compartiments

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.

   Lorsque le tableau de bord s'ouvre, vous pouvez consulter la date la plus récente pour laquelle S3 Storage Lens a collecté des métriques. Votre tableau de bord se charge toujours à la date la plus récente pour laquelle des métriques sont disponibles.
4. Pour voir le classement de vos plus grands compartiments selon la métrique Total storage (Stockage total) pour une plage de dates sélectionnée, faites défiler l'écran jusqu'à la section Top N overview for date (Aperçu des N éléments principaux pour date)

Vous pouvez modifier l'ordre de tri pour afficher les plus petits compartiments. Vous pouvez également ajuster la sélection Metric (Métrique) pour classer vos compartiments en fonction d'une métrique disponible quelconque. La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.

> **Note**
>
> Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

5. Pour obtenir des informations plus détaillées sur vos compartiments, faites défiler la page vers le haut, puis choisissez l'onglet Bucket (Compartiment).

   Dans l'onglet Bucket (Compartiment), vous pouvez voir des détails tels que le taux de croissance récent, la taille moyenne d'objet, les préfixes les plus importants et le nombre d'objets.

Étape 2 : accéder à vos compartiments et mener des investigations

Après avoir identifié vos compartiments S3 les plus importants, vous pouvez accéder à chacun d'eux dans la console S3 pour consulter les objets y figurant, comprendre sa charge de travail associée et identifier ses propriétaires internes. Vous pouvez contacter les propriétaires des compartiments pour déterminer si cette croissance est attendue ou si elle a besoin d'une surveillance et d'un contrôle plus poussés.

Découverte des compartiments Amazon S3 froids

Si vous avez activé S3 Storage Lens advanced metrics (Métriques avancées S3 Storage Lens), vous pouvez utiliser des métriques d'activité pour comprendre à quel point vos compartiments S3 sont froids. Un compartiment « froid » est un compartiment dont on accède plus (ou très rarement) au stockage. Ce manque d'activité indique généralement que l'on n'accède pas fréquemment aux objets du compartiment.

Les métriques d'activité, telles que Get Requests (Demandes Get) et Download Bytes (Octets de chargement) indiquent la fréquence d'accès quotidien à vos compartiments. Pour comprendre la

cohérence du modèle d'accès et pour repérer les compartiments auxquels on n'accède plus du tout, vous pouvez modifier ces données sur plusieurs mois. La métrique Retrieval Rate (Taux d'extraction), qui est calculée en tant qu'octets de téléchargement/Stockage total, indique la proportion de stockage dans un compartiment auquel on accède quotidiennement.

> **Note**
>
> Les octets de téléchargement sont dupliqués dans les cas où le même objet est téléchargé plusieurs fois au cours de la journée.

Prérequis

Pour voir les métriques d'activité dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) S3 Storage Lens, puis sélectionner Activity metrics (Métriques d'activité). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

Étape 1 : identifier les compartiments actifs

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Choisissez l'onglet Bucket (Compartiment) et faites défiler l'écran jusqu'à la section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date).

   Dans la section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date), vous pouvez tracer vos compartiments sur plusieurs dimensions à l'aide de trois métriques quelconques pour représenter les valeurs X-axis (Axe des X), Y-axis (Axe des Y) et Size (Taille) de la bulle.
5. Pour rechercher les compartiments devenus froids, pour X-axis (Axe des X), Y-axis (Axe des Y) et Size (Taille), choisissez les métriques Total storage (Stockage total), % retrieval rate (Taux d'extraction en %) et Average object size (Taille moyenne des objets).
6. Dans la section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date), localisez les compartiments dont le taux d'extraction est égal à zéro (ou proche de zéro)

et ayant une taille de stockage relative plus grande, et choisissez la bulle qui représente le compartiment.

Une case apparaît avec des choix pour afficher des informations plus détaillées. Effectuez l'une des actions suivantes :

a. Pour mettre à jour l'onglet Bucket (Compartiment) afin d'afficher les métriques uniquement pour le compartiment sélectionné, choisissez Drill down (Approfondir), puis choisissez Apply (Appliquer).
b. Pour agréger vos données au niveau du compartiment par compte, Région AWS, classe de stockage ou compartiment, choisissez Analyze by (Analyser par) et effectuez un choix pour Dimension. Par exemple, pour agréger par classe de stockage, choisissez Storage class (Classe de stockage) pour Dimension.

Pour trouver des compartiments qui sont devenus froids, effectuez une analyse à bulles à l'aide des métriques Total storage (Stockage total), % Retrieval Rate (% de taux d'extraction) et Average object size (Taille moyenne des objets). Recherchez les compartiments dont le taux de récupération est égal à zéro (ou proche de zéro) et ayant une taille de stockage relative plus grande.

L'onglet Bucket (Compartiment) de votre tableau de bord est mis à jour pour afficher les données relatives à l'agrégation ou au filtre que vous avez sélectionné(e). Si vous avez agrégé par classe de stockage ou par une autre dimension, ce nouvel onglet s'ouvre dans votre tableau de bord (par exemple, l'onglet Storage class (Classe de stockage)).

Étape 2 : mener des investigations sur les compartiments froids

À ce stade, vous pouvez identifier les propriétaires des compartiments froids dans votre compte ou votre organisation et déterminer si ce stockage est toujours nécessaire. Vous pouvez ensuite optimiser les coûts avec des configurations d'expiration de cycle de vie pour ces compartiments ou en archivant des données dans l'une des classes de stockage Amazon S3 Glacier.

Pour éviter le problème des compartiments froids à l'avenir, vous pouvez effectuer une transition automatique de vos données à l'aide de configurations de cycle de vie S3 pour vos compartiments, ou vous pouvez activer l'archivage automatique avec S3 Intelligent-Tiering.

Vous pouvez également utiliser l'étape 1 pour identifier les compartiments chauds. Ensuite, vous pouvez vous assurer que ces compartiments utilisent la classe de stockage S3 appropriée afin de répondre à leurs demandes de la manière la plus efficace en termes de performances et de coûts.

Localiser les chargements partitionnés incomplets

Vous pouvez utiliser les chargements partitionnés pour charger des objets de très grande taille (jusqu'à 5 To) en tant qu'ensemble de parties pour améliorer le débit et accélérer la récupération suite à des problèmes réseau. Dans les cas où le processus de chargement partitionné ne se termine pas, les parties incomplètes restent dans le compartiment (dans un état inutilisable). Ces parties incomplètes entraînent des coûts de stockage jusqu'à la fin du processus de chargement ou jusqu'à leur suppression. Pour de plus amples informations, veuillez consulter Chargement et copie d'objets à l'aide d'un chargement partitionné.

Avec S3 Storage Lens, vous pouvez identifier le nombre d'octets des chargements partitionnés non terminés dans votre compte ou dans l'ensemble de votre organisation, y compris les chargements partitionnés non terminés datant de plus de 7 jours. Pour obtenir la liste complète des métriques des chargements partitionnés non terminés, consultez Glossaire des métriques Amazon S3 Storage Lens.

À titre de bonne pratique, nous vous recommandons de configurer des règles de cycle de vie pour faire expirer les chargements partitionnés non terminés datant de plus d'un certain nombre de jours. Lorsque vous créez votre règle de cycle de vie pour faire expirer les chargements partitionnés non terminés, nous recommandons un délai de 7 jours comme point de départ.

Étape 1 : passer en revue les tendances générales relatives aux chargements partitionnés non terminés

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Dans la section Snapshot for date (Instantané pour date), sous Metrics categories (Catégories de métriques), choisissez Cost optimization (Optimisation des coûts).

La section Snapshot for date (Instantané pour date) est mise à jour pour afficher les métriques Cost optimization (Optimisation des coûts), qui incluent Incomplete multipart upload bytes greater than 7 days old (Octets de chargements partitionnés non terminés datant de plus de 7 jours).

Dans n'importe quel graphique de votre tableau de bord S3 Storage Lens, vous pouvez consulter les métriques des chargements partitionnés non terminés. Vous pouvez utiliser ces métriques pour évaluer plus en détail l'impact des octets des chargements partitionnés non terminés sur votre stockage, y compris leur contribution aux tendances générales de croissance. Vous pouvez également accéder à des niveaux d'agrégation plus profonds en utilisant les onglets Account (Compte), Région AWS, Bucket (Compartiment) ou Storage class (Classe de stockage) pour une analyse approfondie de vos données. Pour voir un exemple, consultez Découverte des compartiments Amazon S3 froids.

Étape 2 : identifier les compartiments contenant le plus grand nombre d'octets de chargements partitionnés non terminés mais ne disposant pas de règles de cycle de vie permettant d'abandonner les chargements partitionnés non terminés

Prérequis

Pour consulter la métrique Abort incomplete multipart upload lifecycle rule count (Nombre de règles de cycle de vie d'abandon de chargements partitionnés non terminés) dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) S3 Storage Lens, puis sélectionner Advanced cost optimization metrics (Métriques avancées d'optimisation des coûts). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Pour identifier les compartiments spécifiques qui accumulent des chargements partitionnés non terminés datant de plus de 7 jours, accédez à la section Top N overview for date (Aperçu des N éléments principaux pour date).

Par défaut, la section Top N overview for date (Aperçu des N éléments principaux pour date) affiche les métriques des 3 principaux compartiments. Vous pouvez augmenter ou diminuer le nombre de compartiments dans le champ Top N (N éléments principaux). La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. (Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.)

> **Note**
>
> Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

5. Pour Metric (Métrique), choisissez Incomplete multipart upload bytes greater than 7 days old (Octets de chargements partitionnés non terminés datant de plus de 7 jours) dans la catégorie Cost optimization (Optimisation des coûts).

   Sous Top number buckets (nombre compartiments principaux), vous pouvez voir les compartiments contenant le plus d'octets de chargements partitionnés non terminés datant de plus de 7 jours.

6. Pour afficher des métriques plus détaillées au niveau des compartiments pour les chargements partitionnés non terminés, faites défiler l'affichage jusqu'en haut de la page, puis choisissez l'onglet Bucket (Compartiment).

7. Faites défiler jusqu'à la section Buckets (Compartiments). Pour Metrics categories (Catégories de métriques), sélectionnez Cost optimization (Optimisation des coûts). Effacez ensuite Summary (Résumé).

   La liste Buckets (Compartiments) est mise à jour pour afficher toutes les métriques Cost optimization (Optimisation des coûts) disponibles pour les compartiments affichés.

8. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement les métriques sur l'optimisation des coûts spécifiques, choisissez l'icône des préférences ().

9. Désactivez les boutons bascules pour toutes les métriques sur l'optimisation des coûts jusqu'à ce que seules les métriques Incomplete multipart upload bytes greater than 7 days old (Octets de chargements partitionnés non terminés datant de plus de 7 jours) et Abort incomplete multipart

upload lifecycle rule count (Nombre de règles de cycle de vie d'abandon de chargements partitionnés non terminés) restent sélectionnées.

10. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
11. Choisissez Confirm (Confirmer).

    La liste Buckets (Compartiments) est mise à jour pour afficher les métriques au niveau des compartiments pour les chargements partitionnés non terminés et les nombres de règles de cycle de vie. Vous pouvez utiliser ces données pour identifier les compartiments contenant le plus grand nombre d'octets de chargements partitionnés non terminés datant de plus de 7 jours et ne présentant pas de règles de cycle de vie permettant d'abandonner les chargements partitionnés non terminés. Ensuite, vous pouvez accéder à ces compartiments dans la console S3 et ajouter des règles de cycle de vie pour supprimer les chargements partitionnés non terminés abandonnés.

Étape 3 : ajouter une règle de cycle de vie pour supprimer les chargements partitionnés non terminés après 7 jours

Pour gérer automatiquement les chargements partitionnés non terminés, vous pouvez utiliser la console S3 pour créer une configuration de cycle de vie afin de faire expirer les octets de chargements partitionnés non terminés d'un compartiment après un nombre de jours spécifié. Pour de plus amples informations, veuillez consulter Configuration d'une configuration de cycle de vie de compartiment pour supprimer les chargements partitionnés incomplets.

Réduire le nombre de versions anciennes conservées

Lorsque cette option est activée, la fonction de gestion des versions S3 conserve plusieurs copies distinctes du même objet que vous pouvez utiliser pour récupérer rapidement les données si un objet est supprimé ou écrasé accidentellement. Si vous avez activé la gestion des versions S3 sans configurer de règles de cycle de vie pour faire passer ou expirer des versions anciennes, un grand nombre de versions anciennes peuvent s'accumuler, ce qui peut avoir des répercussions sur les coûts de stockage. Pour de plus amples informations, veuillez consulter Utilisation de la gestion des versions dans les compartiments S3.

Étape 1 : identifier les compartiments contenant le plus grand nombre de versions d'objet anciennes

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Dans la section Snapshot for date (Instantané pour date), sous Metric categories (Catégories de métriques), choisissez Cost optimization (Optimisation des coûts).

   La section Snapshot for date (Instantané pour date) est mise à jour pour afficher les métriques Cost optimization (Optimisation des coûts), qui incluent la métrique % noncurrent version bytes (% d'octets de version ancienne). La métrique % noncurrent version bytes (% d'octets de version ancienne) représente la proportion du nombre total d'octets de stockage qui est attribuée à des versions anciennes, dans la portée du tableau de bord et pour la date sélectionnée.

   > **Note**
   >
   > Si votre métrique % noncurrent version bytes (% d'octets de version ancienne) est supérieure à 10 % de votre stockage au niveau du compte, cela peut indiquer que vous stockez trop de versions d'objet.

5. Pour identifier des compartiments spécifiques qui accumulent un grand nombre de versions anciennes :

   a. Faites défiler l'écran vers le bas jusqu'à la section Top N overview for date (Aperçu des N éléments principaux pour date). Pour Top N (N éléments principaux), entrez le nombre de compartiments pour lesquels vous souhaitez voir des données.

   b. Pour Metric (Métrique), choisissez % noncurrent version bytes (% d'octets de version ancienne).

      Sous Top number buckets (nombre compartiments principaux), vous pouvez voir les compartiments (pour le nombre que vous avez spécifié) avec la métrique % noncurrent version bytes (% d'octets de version ancienne) la plus élevée. La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.

> **Note**
>
> Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

c. Pour afficher des métriques plus détaillées au niveau des compartiments pour les versions d'objet anciennes, faites défiler l'affichage jusqu'en haut de la page, puis choisissez l'onglet Bucket (Compartiment).

   Dans un graphique ou une visualisation quelconque de votre tableau de bord S3 Storage Lens, vous pouvez accéder à des niveaux d'agrégation plus approfondis à l'aide des onglets Account (Compte), Région AWS, Storage class (Classe de stockage) ou Bucket (Compartiment). Pour voir un exemple, consultez Découverte des compartiments Amazon S3 froids.

d. Dans la section Buckets (Compartiments), pour Metric categories (Catégories de métriques), sélectionnez Cost optimization (Optimisation des coûts). Effacez ensuite Summary (Résumé).

   Vous pouvez désormais voir la métrique % noncurrent version bytes (% d'octets de version ancienne), ainsi que d'autres métriques relatives aux versions anciennes.

Étape 2 : identifier les compartiments auxquels il manque des règles de cycle de vie de transition et d'expiration pour la gestion des versions anciennes

Prérequis

Pour consulter les métriques Noncurrent version transition lifecycle rule count (Nombre de règles de cycle de vie des transitions de version ancienne) et Noncurrent version expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration de version ancienne) dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) S3 Storage Lens, puis sélectionner Advanced cost optimization metrics (Métriques avancées d'optimisation des coûts). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Dans votre tableau de bord Storage Lens, choisissez l'onglet Bucket (Compartiment).
5. Faites défiler jusqu'à la section Buckets (Compartiments). Pour Metrics categories (Catégories de métriques), sélectionnez Cost optimization (Optimisation des coûts). Effacez ensuite Summary (Résumé).

   La liste Buckets (Compartiments) est mise à jour pour afficher toutes les métriques Cost optimization (Optimisation des coûts) disponibles pour les compartiments affichés.
6. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement les métriques sur l'optimisation des coûts spécifiques, choisissez l'icône des préférences ().
7. Désactivez les boutons bascules pour toutes les métriques sur l'optimisation des coûts jusqu'à ce que seules les métriques suivantes restent sélectionnées :

   - % noncurrent version bytes (% d'octets de version ancienne)
   - Noncurrent version transition lifecycle rule count (Nombre de règles de cycle de vie des transitions de version ancienne)
   - Noncurrent version expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration de version ancienne)
8. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
9. Choisissez Confirm (Confirmer).

   La liste Buckets (Compartiments) est mise à jour pour afficher les métriques relatives aux octets de version ancienne et aux nombres de règles de cycle de vie de version ancienne. Vous pouvez utiliser ces données pour identifier les compartiments qui contiennent un pourcentage élevé d'octets de version ancienne mais auxquels il manque des règles de cycle de vie de transition et d'expiration. Vous pouvez ensuite accéder à ces compartiments dans la console S3 et leur ajouter des règles de cycle de vie.

Étape 3 : ajouter des règles de cycle de vie pour transférer ou faire expirer des versions d'objet anciennes

Une fois que vous avez déterminé quels compartiments doivent être examinés davantage, vous pouvez accéder aux compartiments dans la console S3 et ajouter une politique de cycle de vie pour faire expirer les versions anciennes après un nombre de jours spécifié. Sinon, pour réduire les coûts tout en conservant les versions anciennes, vous pouvez configurer une politique de cycle de vie pour transférer les versions anciennes vers l'une des classes de stockage Amazon S3 Glacier. Pour de plus amples informations, veuillez consulter Exemple 6 : Spécification d'une règle de cycle de vie pour un compartiment activé pour la gestion des versions.

Identification des compartiments dépourvus de règles de cycle de vie et examen des nombres de règles de cycle de vie

S3 Storage Lens fournit des métriques relatives aux nombres de règles de cycle de vie S3 que vous pouvez utiliser pour identifier les compartiments auxquels il manque des règles de cycle de vie. Pour rechercher les compartiments dépourvus de règles de cycle de vie, vous pouvez utiliser la métrique Total buckets without lifecycle rules (Nombre total de compartiments sans règles de cycle de vie). Un compartiment sans configuration de cycle de vie S3 peut contenir un espace de stockage dont vous n'avez plus besoin ou que vous pouvez migrer vers une classe de stockage moins coûteuse. Vous pouvez également utiliser des métriques relatives aux nombres de règles de cycle de vie pour identifier les compartiments auxquels il manque des types spécifiques de règles de cycle de vie, tels que des règles d'expiration ou de transition.

Prérequis

Pour consulter les métriques relatives aux nombres de règles de cycle de vie et la métrique Total buckets without lifecycle rules (Nombre total de compartiments sans règles de cycle de vie) dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) S3 Storage Lens, puis sélectionner Advanced cost optimization metrics (Métriques avancées sur l'optimisation des coûts). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

Étape 1 : identifier les compartiments sans règles de cycle de vie

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).

3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Pour identifier des compartiments spécifiques sans règles de cycle de vie, faites défiler l'affichage vers le bas jusqu'à la section Top N overview for date (Aperçu des N éléments principaux pour date).

   Par défaut, la section Top N overview for date (Aperçu des N éléments principaux pour date) affiche les métriques des 3 principaux compartiments. Dans le champ Top N (N éléments principaux), vous pouvez augmenter le nombre de compartiments. La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.

   > **Note**
   >
   > Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

5. Pour Metric (Métrique), choisissez Total buckets without lifecycle rules (Nombre total de compartiments sans règles de cycle de vie) dans la catégorie Cost optimization (Optimisation des coûts).
6. Consultez les données suivantes pour Total buckets without lifecycle rules (Nombre total de compartiments sans règles de cycle de vie) :

   - Top number accounts (nombre comptes principaux) – Découvrez les comptes qui contiennent le plus de compartiments sans règles de cycle de vie.
   - Top number Regions (nombre régions principales) – Consultez la répartition des compartiments sans règles de cycle de vie par région.
   - Top number buckets (nombre compartiments principaux) – Découvrez quels compartiments sont dépourvus de règles de cycle de vie.

   Dans un graphique ou une visualisation quelconque de votre tableau de bord S3 Storage Lens, vous pouvez accéder à des niveaux d'agrégation plus approfondis à l'aide des onglets Account

(Compte), Région AWS, Storage class (Classe de stockage) ou Bucket (Compartiment). Pour voir un exemple, consultez Découverte des compartiments Amazon S3 froids.

Après avoir identifié les compartiments dépourvus de règles de cycle de vie, vous pouvez également passer en revue les nombres de règles de cycle de vie spécifiques pour vos compartiments.

Étape 2 : passer en revue les nombres de règles de cycle de vie pour vos compartiments

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord que vous souhaitez afficher.
4. Dans votre tableau de bord S3 Storage Lens, choisissez l'onglet Bucket (Compartiment).
5. Faites défiler jusqu'à la section Buckets (Compartiments). Sous Metrics categories (Catégories de métriques), sélectionnez Cost optimization (Optimisation des coûts). Effacez ensuite Summary (Résumé).

   La liste Buckets (Compartiments) est mise à jour pour afficher toutes les métriques Cost optimization (Optimisation des coûts) disponibles pour les compartiments affichés.
6. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement les métriques sur l'optimisation des coûts spécifiques, choisissez l'icône des préférences ().
7. Désactivez les boutons bascules pour toutes les métriques sur l'optimisation des coûts jusqu'à ce que seules les métriques suivantes restent sélectionnées :

   - Transition lifecycle rule count (Nombre de règles de cycle de vie de transition)
   - Expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration)
   - Noncurrent version transition lifecycle rule count (Nombre de règles de cycle de vie des transitions de version ancienne)
   - Noncurrent version expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration de version ancienne)

- Abort incomplete multipart upload lifecycle rule count (Nombre de règles de cycle de vie d'abandon de chargements partitionnés non terminés)
- Total lifecycle rule count (Nombre total de règles de cycle de vie)

8. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
9. Choisissez Confirm (Confirmer).

   La liste Buckets (Compartiments) est mise à jour pour afficher les métriques relatives aux nombres de règles de cycle de vie pour vos compartiments. Vous pouvez utiliser ces données pour identifier les compartiments sans règles de cycle de vie ou les compartiments auxquels il manque certains types de règles de cycle de vie, par exemple des règles d'expiration ou de transition. Vous pouvez ensuite accéder à ces compartiments dans la console S3 et leur ajouter des règles de cycle de vie.

Étape 3 : ajouter des règles de cycle de vie

Après avoir identifié les compartiments sans règles de cycle de vie, vous pouvez ajouter des règles de cycle de vie. Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket et Exemples de configuration de cycle de vie S3.

Utiliser S3 Storage Lens pour protéger vos données

Vous pouvez utiliser les métriques sur la protection des données Amazon S3 Storage Lens pour identifier les compartiments dans lesquels les bonnes pratiques de protection des données n'ont pas été appliquées. Vous pouvez utiliser ces métriques pour agir et appliquer des paramètres standard conformes aux bonnes pratiques pour protéger vos données dans les compartiments de votre compte ou de votre organisation. Par exemple, vous pouvez utiliser des métriques sur la protection des données pour identifier les compartiments qui n'utilisent pas de clés AWS Key Management Service (AWS KMS) (SSE-KMS) pour le chiffrement par défaut ou les demandes qui utilisent AWS Signature Version 2 (SigV2).

Les cas d'utilisation suivants fournissent des stratégies d'utilisation de votre tableau de bord S3 Storage Lens pour identifier les anomalies et appliquer les bonnes pratiques de protection des données sur vos compartiments S3.

Rubriques

- Identification des compartiments qui n'utilisent pas le chiffrement côté serveur avec AWS KMS pour le chiffrement par défaut (SSE-KMS)

- Identification des compartiments pour lesquels la gestion des versions S3 est activée
- Identification des demandes qui utilisent AWS Signature Version 2 (SigV2)
- Décompte du nombre total de règles de réplication pour chaque compartiment
- Identification du pourcentage d'octets de verrouillage d'objets

Identification des compartiments qui n'utilisent pas le chiffrement côté serveur avec AWS KMS pour le chiffrement par défaut (SSE-KMS)

Avec le chiffrement par défaut d'Amazon S3, vous pouvez définir le comportement de chiffrement par défaut pour un compartiment S3. Pour de plus amples informations, veuillez consulter the section called “Définition du chiffrement du compartiment par défaut”.

Vous pouvez utiliser les métriques SSE-KMS enabled bucket count (Nombre de compartiments avec SSE-KMS activé) et % SSE-KMS enabled buckets (% de compartiments avec SSE-KMS activé) pour identifier les compartiments qui utilisent le chiffrement côté serveur à l'aide de clés AWS KMS (SSE-KMS) pour le chiffrement par défaut. S3 Storage Lens fournit également des métriques pour les octets non chiffrés, les objets non chiffrés, les octets chiffrés et les objets chiffrés. Pour obtenir une liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.

Vous pouvez analyser les métriques de chiffrement SSE-KMS dans le contexte des métriques de chiffrement générales afin d'identifier les compartiments qui n'utilisent pas SSE-KMS. Pour utiliser SSE-KMS pour tous les compartiments de votre compte ou de votre organisation, vous pouvez ensuite mettre à jour les paramètres de chiffrement par défaut de ces compartiments afin d'utiliser SSE-KMS. Outre SSE-KMS, vous pouvez utiliser le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) ou des clés fournies par le client (SSE-C). Pour de plus amples informations, veuillez consulter Protection des données à l'aide du chiffrement.

Étape 1 : identifier les compartiments qui utilisent SSE-KMS pour le chiffrement par défaut

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.

4. Dans la section Trends and distributions (Tendances et distributions), choisissez % SSE-KMS enabled buckets (% de compartiments avec SSE-KMS activé) comme métrique principale et % encrypted bytes (% d'octets chiffrés) comme métrique secondaire.

   Le graphique Trend for date (Tendance pour date) est mis à jour pour afficher les tendances relatives à SSE-KMS et aux octets chiffrés.

5. Pour afficher des informations plus détaillées au niveau du compartiment pour SSE-KMS :

   a. Choisissez un point sur le graphique. Une case apparaît avec des choix pour afficher des informations plus détaillées.

   b. Choisissez la dimension Buckets (Compartiments). Choisissez ensuite Apply(Applisuer).

6. Dans le graphique Distribution by buckets for date (Distribution par compartiments pour date), choisissez la métrique SSE-KMS enabled bucket count (Nombre de compartiments avec SSE-KMS activé).

7. Vous pouvez désormais voir pour quels compartiments SSE-KMS est activé et pour quels compartiments il ne l'est pas.

Étape 2 : mettre à jour les paramètres de chiffrement par défaut des compartiments

Maintenant que vous avez déterminé quels compartiments utilisent SSE-KMS dans le contexte de votre métrique % encrypted bytes (% d'octets chiffrés), vous pouvez identifier les compartiments qui n'utilisent pas SSE-KMS. Vous pouvez ensuite éventuellement accéder à ces compartiments dans la console S3 et mettre à jour leurs paramètres de chiffrement par défaut pour utiliser SSE-KMS ou SSE-S3. Pour de plus amples informations, veuillez consulter Configuration du chiffrement par défaut.

Identification des compartiments pour lesquels la gestion des versions S3 est activée

Lorsque cette option est activée, la fonction de gestion des versions S3 conserve plusieurs versions du même objet qui peuvent être utilisées pour récupérer rapidement des données si un objet est supprimé ou écrasé accidentellement. Vous pouvez utiliser la métrique Versioning-enabled bucket count (Nombre de compartiments activés pour la gestion des versions) pour voir quels compartiments utilisent la gestion des versions S3. Ensuite, vous pouvez agir dans la console S3 pour activer la gestion des versions S3 pour d'autres compartiments.

Étape 1 : identifier les compartiments pour lesquels la gestion des versions S3 est activée

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le volet de navigation, sélectionnez Storage Lens, puis Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans la section Trends and distributions (Tendances et distributions), choisissez Versioning-enabled bucket count (Nombre de compartiments activés pour la gestion des versions) comme métrique principale et Buckets (Compartiments) comme métrique secondaire.

   Le graphique Trend for date (Tendance pour date) est mis à jour pour afficher les tendances des compartiments pour lesquels la gestion des versions S3 est activée. Juste en dessous de la courbe des tendances, vous pouvez voir les sous-sections Storage class distribution (Distribution des classes de stockage) et Region distribution (Distribution par région).
5. Pour obtenir des informations plus détaillées sur l'un des compartiments que vous voyez dans le graphique Trend for date (Tendance pour date) afin de pouvoir effectuer une analyse approfondie, procédez comme suit :

   a. Choisissez un point sur le graphique. Une case apparaît avec des choix pour afficher des informations plus détaillées.

   b. Choisissez une dimension à appliquer à vos données pour une analyse plus approfondie : Account (Compte), Région AWS, Storage class (Classe de stockage) ou Bucket (Compartiment). Choisissez ensuite Apply(Applisuer).
6. Dans la section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date), choisissez les métriques Versioning-enabled bucket count (Nombre de compartiments activés pour la gestion des versions), Buckets (Compartiments) et Active buckets (Compartiments actifs).

   La section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date) est mise à jour pour afficher les données relatives aux métriques que vous avez sélectionnées. Vous pouvez utiliser ces données pour voir pour quels compartiments la gestion des versions S3 est activée dans le contexte de votre nombre total de compartiments. Dans la section Bubble analysis by buckets for date (Analyse à bulles par compartiments pour date), vous pouvez tracer vos compartiments sur plusieurs dimensions à l'aide de trois métriques quelconques pour représenter les valeurs X-axis (Axe des X), Y-axis (Axe des Y) et Size (Taille) de la bulle.

Étape 2 : activer la gestion des versions S3

Après avoir identifié les compartiments pour lesquels la gestion des versions S3 est activée, vous pouvez identifier les compartiments pour lesquels la gestion des versions S3 n'a jamais été activée ou a été suspendue. Ensuite, vous pouvez éventuellement activer la gestion des versions pour ces compartiments dans la console S3. Pour de plus amples informations, veuillez consulter Activation de la gestion des versions sur les compartiments.

Identification des demandes qui utilisent AWS Signature Version 2 (SigV2)

Vous pouvez utiliser la métrique All unsupported signature requests (Toutes les demandes de signature non prises en charge) pour identifier les demandes qui utilisent AWS Signature Version 2 (SigV2). Ces données peuvent vous aider à identifier les applications spécifiques qui utilisent SigV2. Vous pouvez ensuite migrer ces applications vers AWS Signature Version 4 (SigV4).

SigV4 est la méthode de signature recommandée pour toutes les nouvelles applications S3. SigV4 offre une sécurité améliorée et est pris en charge dans toutes les Régions AWS. Pour plus d'informations, consultez Amazon S3 update - SigV2 deprecation period extended & modified (Mise à jour Amazon S3 – Période d'obsolescence SigV2 étendue et modifiée).

Prérequis

Pour voir la métrique All unsupported signature requests (Toutes les demandes de signature non prises en charge) dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) de S3 Storage Lens, puis sélectionner Advanced data protection metrics (Métriques avancées sur la protection des données). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

Étape 1 : examiner les tendances de signature SIGv2 par Compte AWS, région et compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Pour identifier des compartiments, des comptes et des régions spécifiques dont les demandes utilisent SigV2 :

a. Sous Top N overview for date (Aperçu des N éléments principaux pour date), dans Top N (N éléments principaux), entrez le nombre de compartiments pour lesquels vous souhaitez voir des données.

b. Pour Metric (Métrique), choisissez All unsupported signature requests (Toutes les demandes de signature non prises en charge) dans la catégorie Data protection (Protection des données).

   La métrique Top N overview for date (Aperçu des N éléments principaux pour date) est mise à jour pour afficher les données des demandes SigV2 par compte, Région AWS et compartiment. La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.

   > **Note**
   >
   > Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

Étape 2 : identifier les compartiments auxquels les applications accèdent via des demandes SigV2

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans votre tableau de bord Storage Lens, choisissez l'onglet Bucket (Compartiment).
5. Faites défiler jusqu'à la section Buckets (Compartiments). Sous Metrics categories (Catégories de métriques), choisissez Data protection (Protection des données). Effacez ensuite Summary (Résumé).

La liste Buckets (Compartiments) est mise à jour pour afficher toutes les métriques Data protection (Protection des données) disponibles pour les compartiments affichés.

6. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement des métriques spécifiques sur la protection des données, choisissez l'icône des préférences ().
7. Désactivez les boutons bascules pour toutes les métriques sur la protection des données jusqu'à ce que seules les métriques suivantes restent sélectionnées :
   - All unsupported signature requests (Toutes les demandes de signature non prises en charge)
   - % all unsupported signature requests (% de toutes les demandes de signature non prises en charge)
8. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
9. Choisissez Confirm (Confirmer).

   La liste Buckets (Compartiments) est mise à jour pour afficher les métriques au niveau des compartiments pour les demandes SigV2. Vous pouvez utiliser ces données pour identifier des compartiments spécifiques qui font l'objet de demandes SigV2. Vous pouvez ensuite utiliser ces informations pour migrer vos applications vers SigV4. Pour de plus amples informations, veuillez consulter Demandes d'authentification (AWS Signature Version 4) dans la Référence API Amazon Simple Storage Service.

Décompte du nombre total de règles de réplication pour chaque compartiment

La réplication S3 permet la copie automatique et asynchrone d'objets entre les compartiments Amazon S3. Les compartiments configurés pour la réplication d'objet peuvent appartenir au même Compte AWS ou à des comptes distincts. Pour de plus amples informations, veuillez consulter Vue d'ensemble de la réplication d'objets.

Vous pouvez utiliser les métriques de décompte des règles de réplication S3 Storage Lens pour obtenir des informations détaillées par compartiment sur les compartiments configurés pour la réplication. Ces informations incluent les règles de réplication au sein des compartiments et des régions et entre eux.

Prérequis

Pour voir les métriques de décompte des règles de réplication dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) de S3 Storage Lens, puis sélectionner Advanced data protection metrics (Métriques avancées sur la protection des données). Pour de plus amples informations, veuillez consulter Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

Étape 1 : Compter le nombre total de règles de réplication pour chaque compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans votre tableau de bord Storage Lens, choisissez l'onglet Bucket (Compartiment).
5. Faites défiler jusqu'à la section Buckets (Compartiments). Sous Metrics categories (Catégories de métriques), choisissez Data protection (Protection des données). Effacez ensuite Summary (Résumé).
6. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement les métriques de décompte des règles de réplication, choisissez l'icône des préférences ().
7. Désactivez les boutons bascules pour toutes les métriques sur la protection des données jusqu'à ce que seules les métriques de décompte des règles de réplication restent sélectionnées :
   - Same-Region Replication rule count (Nombre de règles de réplication pour la même région)
   - Cross-Region Replication rule count (Nombre de règles de réplication entre régions)
   - Same-account replication rule count (Nombre de règles de réplication pour le même compte)
   - Cross-account replication rule count (Nombre de règles de réplication entre comptes)
   - Total replication rule count (Nombre total de règles de réplication)
8. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
9. Choisissez Confirm (Confirmer).

Étape 2 : ajouter des règles de réplication

Une fois que vous avez déterminé le nombre de règles de réplication par compartiment, vous pouvez éventuellement créer des règles de réplication supplémentaires. Pour de plus amples informations, veuillez consulter Exemples de configuration de la réplication en direct.

Identification du pourcentage d'octets de verrouillage d'objets

Avec le verrouillage des objets S3, vous pouvez stocker des objets selon un modèle Write Once Read Many (WORM). Vous pouvez utiliser le verrouillage des objets pour empêcher que des objets soient supprimés ou remplacés pendant une durée déterminée ou indéfinie. Vous pouvez activer le verrouillage des objets uniquement lorsque vous créez un compartiment et activez également la gestion des versions S3. Toutefois, vous pouvez modifier la période de conservation pour des versions d'objet individuelles ou appliquer des restrictions légales aux compartiments pour lesquels le verrouillage des objets est activé. Pour de plus amples informations, veuillez consulter Utilisation du verrouillage des objets S3.

Vous pouvez utiliser les métriques de verrouillage d'objets dans S3 Storage Lens pour voir la métrique % Object Lock bytes (% d'octets de verrouillage d'objets) pour votre compte ou votre organisation. Vous pouvez utiliser ces informations pour identifier les compartiments de votre compte ou de votre organisation qui ne respectent pas vos bonnes pratiques de protection des données.

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans la section Snapshot (Instantané), sous Metrics categories (Catégories de métriques), choisissez Data protection (Protection des données).

   La section Snapshot (Instantané) est mise à jour pour afficher les métriques sur la protection des données, notamment la métrique % Object Lock bytes (% d'octets de verrouillage d'objets). Vous pouvez voir le pourcentage global d'octets de verrouillage d'objets pour votre compte ou votre organisation.
5. Pour voir la métrique % Object Lock bytes (% d'octets de verrouillage d'objets) par compartiment, faites défiler la page vers le bas jusqu'à la section Top N overview (Aperçu des N éléments principaux).

Pour obtenir des données au niveau des objets pour le verrouillage d'objets, vous pouvez également utiliser les métriques Object Lock object count (Nombre d'objets de verrouillage d'objets) et % Object Lock objects (% d'objets avec verrouillage d'objets).

6. Pour Metric (Métrique), choisissez % Object Lock bytes (% d'octets de verrouillage d'objets) dans la catégorie Data protection (Protection des données).

   Par défaut, la section Top N overview for date (Aperçu des N éléments principaux pour date) affiche les métriques des 3 principaux compartiments. Dans le champ Top N (N éléments principaux), vous pouvez augmenter le nombre de compartiments. La section Top N overview for date (Aperçu des N éléments principaux pour date) indique également la variation en pourcentage par rapport à la journée ou à la semaine précédente et une ligne étincelante pour visualiser la tendance. Cette tendance est une tendance sur 14 jours pour les métriques gratuites et une tendance sur 30 jours pour les métriques et recommandations avancées.

   > **Note**
   >
   > Grâce aux métriques et recommandations avancées S3 Storage Lens, les métriques sont disponibles pour les requêtes pendant 15 mois. Pour de plus amples informations, veuillez consulter Sélection des métriques.

7. Passez en revue les données suivantes pour % Object Lock bytes (% d'octets de verrouillage d'objets) :

   - Top number accounts (nombre comptes principaux) - Découvrez quels comptes ont la métrique % Object Lock bytes (% d'octets de verrouillage d'objets) la plus élevée et la plus faible.
   - Top number Regions (nombre régions principales) - Affichez une répartition de la métrique % Object Lock bytes (% d'octets de verrouillage d'objets) par région.
   - Top number buckets (nombre compartiments principaux) - Découvrez quels compartiments ont la métrique % Object Lock bytes (% d'octets de verrouillage d'objets) la plus élevée et la plus faible.

Utilisation de S3 Storage Lens pour auditer les paramètres de propriété d'objets

La propriété d'objets S3 est un paramètre au niveau des compartiments S3 que vous pouvez utiliser pour désactiver les listes de contrôle d'accès (ACL) et contrôler la propriété des objets dans

votre compartiment. Si vous définissez la propriété d'objets sur Appliqué par le propriétaire du compartiment, vous pouvez désactiver les listes de contrôle d'accès (ACL) et prendre possession de tous les objets dans votre compartiment. Cette approche simplifie la gestion des accès pour les données stockées dans Amazon S3.

Par défaut, lorsqu'un autre Compte AWS télécharge un objet dans votre compartiment S3, ce compte (le rédacteur d'objet) est propriétaire de l'objet, y a accès et peut en accorder l'accès à d'autres utilisateurs via des ACL. Vous pouvez utiliser la propriété de l'objet pour modifier ce comportement par défaut.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons donc de désactiver les listes ACL, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès de chaque objet individuellement. En définissant la propriété d'objets sur Appliqué par le propriétaire du compartiment, vous pouvez désactiver les listes ACL et vous fier aux politiques pour le contrôle des accès. Pour de plus amples informations, veuillez consulter Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

Avec les métriques de gestion des accès S3 Storage Lens, vous pouvez identifier les compartiments pour lesquels les listes ACL ne sont pas désactivées. Après avoir identifié ces compartiments, vous pouvez migrer les autorisations ACL vers des politiques et désactiver les listes ACL pour ces compartiments.

Rubriques

- Étape 1 : identifier les tendances générales relatives aux paramètres de propriété d'objets
- Étape 2 : identifier les tendances au niveau du compartiment relatives aux paramètres de propriété d'objets
- Étape 3 : mettre à jour votre paramètre de propriété d'objets sur Appliqué par le propriétaire du compartiment afin de désactiver les listes ACL

Étape 1 : identifier les tendances générales relatives aux paramètres de propriété d'objets

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).

3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans la section Snapshot for date (Instantané pour date), sous Metrics categories (Catégories de métriques), choisissez Access management (Gestion des accès).

   La section Snapshot for date (Instantané pour date) est mise à jour pour afficher la métrique % Object Ownership bucket owner enforced (% de compartiments avec propriété d'objets Appliqué par le propriétaire du compartiment). Vous pouvez afficher le pourcentage global de compartiments dans votre compte ou votre organisation qui utilisent le paramètre de propriété d'objets Appliqué par le propriétaire du compartiment, afin de désactiver les listes ACL.

Étape 2 : identifier les tendances au niveau du compartiment relatives aux paramètres de propriété d'objets

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Pour afficher des métriques plus détaillées au niveau du compartiment, choisissez l'onglet Bucket (Compartiment).
5. Dans la section Distribution by buckets for date (Distribution par compartiments pour date), choisissez la métrique % Object Ownership bucket owner enforced (% de compartiments avec propriété d'objets Appliqué par le propriétaire du compartiment).

   Le graphique est mis à jour pour montrer la répartition par compartiment pour % Object Ownership bucket owner enforced (% de compartiments avec propriété d'objets Appliqué par le propriétaire du compartiment). Vous pouvez voir quels compartiments utilisent le paramètre de propriété d'objets Appliqué par le propriétaire du compartiment afin de désactiver les listes ACL.
6. Pour voir les paramètres Appliqué par le propriétaire du compartiment en contexte, faites défiler l'affichage jusqu'à la section Buckets (Compartiments). Pour Metrics categories (Catégories de métriques), sélectionnez Access management (Gestion des accès). Effacez ensuite Summary (Résumé).

La liste Buckets (Compartiments) affiche les données relatives aux trois paramètres de propriété d'objets : Appliqué par le propriétaire du compartiment, Préféré par le propriétaire du compartiment et Enregistreur d'objets.

7. Pour filtrer la liste Buckets (Compartiments) afin d'afficher des métriques uniquement pour un paramètre de propriété d'objets spécifique, choisissez l'icône des préférences ().
8. Effacez les métriques que vous ne souhaitez pas afficher.
9. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
10. Choisissez Confirm (Confirmer).

Étape 3 : mettre à jour votre paramètre de propriété d'objets sur Appliqué par le propriétaire du compartiment afin de désactiver les listes ACL

Après avoir identifié les compartiments qui utilisent le paramètre de propriété d'objets Enregistreur d'objets ou Préféré par le propriétaire du compartiment, vous pouvez migrer vos autorisations ACL vers des politiques de compartiments. Lorsque vous avez terminé la migration de vos autorisations ACL, vous pouvez mettre à jour vos paramètres de propriété d'objets sur Appliqué par le propriétaire du compartiment, afin de désactiver les listes ACL. Pour de plus amples informations, veuillez consulter Conditions préalables à la désactivation des listes ACL.

Utilisation de métriques S3 Storage Lens pour améliorer les performances

Si vous avez activé S3 Storage Lens advanced metrics (Métriques avancées S3 Storage Lens), vous pouvez utiliser des métriques de codes de statut détaillés pour obtenir le nombre de demandes ayant réussi ou échoué. Vous pouvez utiliser ces informations pour résoudre les problèmes d'accès et de performances. Les métriques de codes de statut détaillés indiquent les nombres de codes de statut HTTP, tels que 403 Interdit et 503 Service non disponible. Vous pouvez examiner les tendances globales pour obtenir les métriques de codes de statut détaillés pour les compartiments, les comptes et les organisations S3. Vous pouvez ensuite explorer les métriques au niveau des compartiments afin d'identifier les charges de travail qui accèdent actuellement à ces compartiments et provoquent des erreurs.

Par exemple, vous pouvez consulter la métrique 403 Forbidden error count (Nombre d'erreurs 403 Interdit) pour identifier les charges de travail qui accèdent à des compartiments sans appliquer

les autorisations appropriées. Après avoir identifié ces charges de travail, vous pouvez effectuer un examen détaillé en dehors de S3 Storage Lens pour résoudre vos erreurs 403 Interdit.

Cet exemple montre comment effectuer une analyse de tendances pour l'erreur 403 Interdit à l'aide des métriques 403 Forbidden error count (Nombre d'erreurs 403 Interdit) et % 403 Forbidden errors (% d'erreurs 403 Interdit). Vous pouvez utiliser ces métriques pour identifier les charges de travail qui accèdent aux compartiments sans appliquer les autorisations appropriées. Vous pouvez effectuer une analyse de tendances similaire pour n'importe laquelle des autres métriques de codes de statut détaillés. Pour plus d'informations, consultez Glossaire des métriques Amazon S3 Storage Lens.

Prérequis

Pour voir les métriques de codes de statut détaillés dans votre tableau de bord S3 Storage Lens, vous devez activer Advanced metrics and recommendations (Métriques et recommandations avancées) S3 Storage Lens, puis sélectionner Detailed status code metrics (Métriques de codes de statut détaillés). Pour plus d'informations, consultez Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens.

Rubriques

- Étape 1 : réaliser une analyse des tendances pour un code de statut HTTP individuel
- Étape 2 : analyser les nombres d'erreurs par compartiment
- Étape 3 : entreprendre la résolution des erreurs

Étape 1 : réaliser une analyse des tendances pour un code de statut HTTP individuel

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans la section Trends and distributions (Tendances et distributions), pour Primary metric (Métrique principale), choisissez 403 Forbidden error count (Nombre d'erreurs 403 Interdit) dans la catégorie Detailed status codes (Codes de statut détaillés). Pour Secondary metric (Métrique secondaire), choisissez % 403 Forbidden errors (% d'erreurs 403 Interdit).
5. Faites défiler l'écran vers le bas jusqu'à la section Top N overview for date (Aperçu des N éléments principaux pour date). Pour Metrics (Métriques), choisissez 403 Forbidden error count

(Nombre d'erreurs 403 Interdit) ou % 403 Forbidden errors (% d'erreurs 403 Interdit) dans la catégorie Detailed status codes (Codes de statut détaillés).

La section Top N overview for date (Aperçu des N éléments principaux pour date) est mise à jour pour afficher les nombres d'erreurs 403 Interdit les plus fréquentes par compte, Région AWS et compartiment.

Étape 2 : analyser les nombres d'erreurs par compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), choisissez le nom du tableau de bord que vous souhaitez afficher.
4. Dans votre tableau de bord Storage Lens, choisissez l'onglet Bucket (Compartiment).
5. Faites défiler jusqu'à la section Buckets (Compartiments). Pour Metrics categories (Catégories de métriques), sélectionnez les métriques Detailed status code (Code de statut détaillé). Effacez ensuite Summary (Résumé).

   La liste Buckets (Compartiments) est mise à jour pour afficher toutes les métriques de code de statut détaillées disponibles. Vous pouvez utiliser ces informations pour voir quels compartiments contiennent une grande partie de certains codes de statut HTTP et quels codes de statut sont communs à tous les compartiments.
6. Pour filtrer la liste Buckets (Compartiments) afin d'afficher uniquement des métriques de codes de statut détaillés spécifiques, choisissez l'icône des préférences ().
7. Désélectionnez les boutons bascules correspondant à toutes les métriques de codes de statut détaillés que vous ne souhaitez pas afficher dans la liste Buckets (Compartiments).
8. (Facultatif) Sous Page size (Taille de page), choisissez le nombre de compartiments à afficher dans la liste.
9. Choisissez Confirm (Confirmer).

   La liste Buckets (Compartiments) affiche les métriques des nombres d'erreurs pour le nombre de compartiments que vous avez spécifié. Vous pouvez utiliser ces informations pour identifier les

compartiments spécifiques qui rencontrent de nombreuses erreurs et entreprendre la résolution des erreurs par compartiment.

Étape 3 : entreprendre la résolution des erreurs

Après avoir identifié les compartiments présentant une forte proportion de codes de statut HTTP spécifiques, vous pouvez entreprendre la résolution de ces erreurs. Pour plus d'informations, consultez les ressources suivantes :

- Pourquoi est-ce que j'obtiens l'erreur 403 Interdit quand j'essaie de charger des fichiers dans Amazon S3 ?
- Pourquoi est-ce que j'obtiens l'erreur 403 Interdit quand j'essaie de modifier une politique de compartiments dans Amazon S3 ?
- Comment résoudre les erreurs 403 Interdit provenant de mon compartiment Amazon S3 dont toutes les ressources proviennent du même Compte AWS ?
- Comment résoudre une erreur HTTP 500 ou 503 provenant d'Amazon S3 ?

Glossaire des métriques Amazon S3 Storage Lens

Le glossaire des métriques Amazon S3 Storage Lens fournit la liste complète des métriques gratuites et avancées pour S3 Storage Lens.

S3 Storage Lens offre des métriques gratuites pour tous les tableaux de bord et configurations, avec la possibilité d'une mise à niveau vers les métriques avancées.

- Les métriques gratuites contiennent des métriques pertinentes pour l'utilisation de votre stockage, telles que le nombre de compartiments et les objets dans votre compte. Les métriques gratuites incluent également des métriques basées sur des cas d'utilisation, telles que les métriques sur l'optimisation des coûts et la protection des données. Toutes les métriques gratuites sont collectées quotidiennement et les données sont disponibles pour les requêtes jusqu'à 14 jours.
- Les métriques et recommandations avancées incluent toutes les métriques gratuites et des métriques supplémentaires, telles que les métriques avancées sur la protection des données et l'optimisation des coûts. Les métriques avancées incluent également des catégories de métriques supplémentaires, telles que les métriques d'activité et les métriques de codes de statut détaillées. Les métriques avancées sont disponibles pour les requêtes pendant 15 mois.

Des frais supplémentaires sont facturés lorsque vous utilisez S3 Storage Lens avec des métriques et recommandations avancées. Pour de plus amples informations, consultez la tarification Amazon S3. Pour plus d'informations sur les fonctionnalités des métriques et recommandations avancées, consultez Sélection des métriques.

> **Note**
>
> Pour les groupes Storage Lens, seules les métriques de stockage du niveau d'offre gratuite sont disponibles. Les métriques de niveau avancé ne sont pas disponibles au niveau du groupe Storage Lens.

Noms des métriques

La colonne Nom de la métrique du tableau suivant fournit le nom de chaque métrique de S3 Storage Lens dans la console S3. La colonne CloudWatch et exportation fournit le nom de chaque métrique dans Amazon CloudWatch et le fichier d'exportation de métriques que vous pouvez configurer dans votre tableau de bord S3 Storage Lens.

Formules de métriques dérivées

Les métriques dérivées ne sont pas disponibles pour l'option d'exportation de métriques et de publication sur CloudWatch. Cependant, vous pouvez utiliser les formules de métriques affichées dans la colonne Formule de métrique dérivée pour les calculer.

Interprétation des symboles de préfixe Amazon S3 Storage Lens des multiples des unités de métriques (K, M, G, etc.)

Les multiples des unités des métriques S3 Storage Lens sont écrits avec des symboles de préfixe. Ces symboles de préfixe correspondent aux symboles du Système international d'unités (SI) normalisés par le Bureau international des poids et mesures (BIPM). Ces symboles sont également utilisés dans le Code unifié des unités de mesure (UCUM). Pour plus d'informations, consultez Liste des symboles de préfixe SI.

> **Note**
>
> - L'unité de mesure des octets de stockage S3 est le gigaoctet binaire (Go), où 1 Go correspond à 2^{30} octets, 1 To à 2^{40} octets et 1 Po à 2^{50} octets. Cette unité de mesure est

également connue sous le nom de gibioctet (GiB), selon la définition de la Commission électrotechnique internationale (CEI).

- Lorsqu'un objet est en fin de vie selon la configuration de son cycle de vie, Amazon S3 le place dans une file d'attente en vue de sa suppression et le supprime de manière asynchrone. Cependant, un certain retard est possible entre la date d'expiration et la date à laquelle Amazon S3 supprime l'objet. S3 Storage Lens n'inclut pas de métriques pour les objets qui ont expiré mais qui n'ont pas été supprimés. Pour plus d'informations sur les actions d'expiration dans le cycle de vie S3, consultez Objets en cours d'expiration.

Glossaire des métriques S3 Storage Lens

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Total storage (Stockage total)	StorageBytes	Stockage total, y compris les chargements partitionnés incomplets, les métadonnées des objets et les marqueurs de suppression	Free	Réca atif	N	-
Object count (Nombre d'objets)	ObjectCount	Nombre total d'objets	Free	Réca atif	N	-
Average object size (Taille moyenne des objets)	-	Taille moyenne des objets	Free	Réca atif	Y	sum(Stora geBytes)/ sum(Objec tCount)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Active buckets (Comparti ments actifs)	-	Nombre total de compartiments activement utilisés disposant d'un stockage > 0 octets	Free	Réca atif	Y	-
Compartiments	-	Nombre total de compartiments	Free	Réca atif	Y	-
Comptes	-	Nombre de comptes dont le stockage est dans la portée	Free	Réca atif	Y	-
Current version bytes (Octets de version actuelle)	CurrentVe rsionStor ageBytes	Nombre d'octets qui sont une version actuelle d'un objet	Free	Opti ion des coût	N	-
% current version bytes (% d'octets de version actuelle)	-	Pourcentage d'octets dans la portée qui sont des versions actuelles d'objets	Free	Opti ion des coût	Y	sum(Curre ntVersion StorageBy tes)/ sum(StorageBy tes)
Current version object count (Nombre d'objets de version actuelle)	CurrentVe rsionObje ctCount	Nombre d'objets d'une version actuelle	Free	Opti ion des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% current version objects (% d'objets de version actuelle)	-	Pourcentage d'objets dans l'étendue qui correspondent à une version actuelle	Free	Optii ion des coût	Y	sum(Curre ntVersion ObjectCou nt)/ sum(O bjectCoun t)
Noncurren t version bytes (Octets de version ancienne)	NonCurren tVersionS torageBytes	Nombre d'octets de version ancienne	Free	Optii ion des coût	N	-
% noncurrent version bytes (% d'octets de version ancienne)	-	Pourcentage d'octets dans la portée qui sont des versions non actuelles	Free	Optii ion des coût	Y	sum(NonCu rrentVers ionStorag eBytes)/ s um(Storag eBytes)
Noncurrent version object count (Nombre d'objets de version ancienne)	NonCurren tVersionO bjectCount	Nombre des versions d'objet anciennes	Free	Optii ion des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% noncurrent version objects (% d'objets de version ancienne)	-	Pourcentage d'objets dans la portée d'une version non actuelle	Free	Optii ion des coût	Y	sum(NonCu rrentVers ionObject Count)/ su m(ObjectC ount)
Delete marker bytes (Octets de marqueur de suppression)	DeleteMar kerStorag eBytes	Nombre d'octets dans la portée qui sont des marqueurs de suppression	Free	Optii ion des coût	N	-
% delete marker bytes (% d'octets de marqueur de suppression)	-	Pourcentage d'octets dans la portée qui sont des marqueurs de suppression	Free	Optii ion des coût	Y	sum(Delet eMarkerSt orageByte s)/ sum(St orageByte s)
Delete marker object count (Nombre d'objets marqueur de suppression)	DeleteMar kerObjectCount	Nombre total d'objets avec un marqueur de suppression	Free	Optii ion des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% delete marker objects (% d'objets marqueur de suppression)	-	Pourcentage d'objets dans la portée avec un marqueur de suppression	Free	Opti ion des coût	Y	sum(Delet eMarkerOb jectCount)/ sum(Obj ectCount)
Incomplet e multipart upload bytes (Octets de chargements partitionnés non terminés)	Incomplet eMultipar tUploadSt orageBytes	Nombre total d'octets dans la portée pour des chargements partitionnés non terminés	Free	Opti ion des coût	N	-
% incomplet e multipart upload bytes (% d'octets de chargements partitionnés non terminés)	-	Pourcentage d'octets dans la portée qui sont le résultat de chargements partitionnés non terminés	Free	Opti ion des coût	Y	sum(Incom pleteMult ipartUplo adStorage Bytes)/ su m(Storage Bytes)
Incomplet e multipart upload object count (Nombre d'objets de chargements partitionnés non terminés)	Incomplet eMultipar tUploadOb jectCount	Nombre d'objets dans la portée qui sont des télécharg ements partition nés incomplets	Free	Opti ion des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
% incomplete multipart upload objects (% d'objets de chargements partitionnés non terminés)	-	Pourcentage d'objets dans la portée qui sont des télécharg ements partition nés incomplets	Free	Optii ion des coût	Y	sum(IncompleteMultipartUploadObjectCount)/sum(ObjectCount)
Nombre d'octets de stockage de chargements partitionnés non terminés datant de plus de 7 jours	IncompleteMPUStorageBytesOlderThan7Days	Nombre total d'octets dans la portée pour des chargements partitionnés non terminés datant de plus de 7 jours	Free	Optii ion des coût	N	-
% incomplete multipart upload storage bytes greater than 7 days old (% d'octets de stockage de chargements partitionnés non terminés datant de plus de 7 jours)	-	Pourcentage d'octets pour des chargements partitionnés non terminés datant de plus de 7 jours	Free	Optii ion des coût	Y	sum(IncompleteMPUStorageBytesOlderThan7Days)/sum(StorageBytes)

Nom de la métrique	CloudWatch et exportation	Description	Nivea	Caté[2]	D	Form de métri dériv
Incomplete multipart upload object count greater than 7 days old (Nombre d'objets de chargements partitionnés non terminés datant de plus de 7 jours)	IncompleteMPUObjectCountOlderThan7Days	Nombre d'objets qui sont des chargements partitionnés non terminés datant de plus de 7 jours	Free	Option des coût	N	-
% incomplete multipart upload object count greater than 7 days old (% d'objets de chargements partitionnés non terminés datant de plus de 7 jours)	-	Pourcentage d'objets qui sont des chargements partitionnés non terminés datant de plus de 7 jours	Free	Option des coût	Y	sum(IncompleteMPUObjectCountOlderThan7Days)/sum(ObjectCount)
Transition lifecycle rule count (Nombre de règles de cycle de vie de transition)	TransitionLifecycleRuleCount	Nombre de règles de cycle de vie permettant de transférer des objets vers une autre classe de stockage	Avan	Option des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Average transition lifecycle rules per bucket (Nombre moyen de règles de cycle de vie de transition par compartiment)	-	Nombre moyen de règles de cycle de vie permettant de transférer des objets vers une autre classe de stockage	Avan	Opti ion des coût	Y	sum(Trans itionLife cycleRule Count)/ su m(Distinc tNumberOf Buckets)
Expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration)	Expiratio nLifecycl eRuleCount	Nombre de règles de cycle de vie servant à faire expirer les objets	Avan	Opti ion des coût	N	-
Average expiration lifecycle rules per bucket (Nombre moyen de règles de cycle de vie d'expiration par compartiment)	-	Nombre moyen de règles de cycle de vie servant à faire expirer les objets	Avan	Opti ion des coût	Y	sum(Expir ationLife cycleRule Count)/ su m(Distinc tNumberOf Buckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Noncurrent version transition lifecycle rule count (Nombre de règles de cycle de vie des transitions de version ancienne)	NoncurrentVersionTransitionLifecycleRuleCount	Nombre de règles de cycle de vie permettant de transférer des versions d'objet anciennes vers une autre classe de stockage	Avan	Optim ion des coût	N	
Average noncurrent version transition lifecycle rules per bucket (Nombre moyen de règles de cycle de vie de transition de version ancienne par compartiment)	-	Nombre moyen de règles de cycle de vie permettant de transférer des versions d'objet anciennes vers une autre classe de stockage	Avan	Optim ion des coût	Y	sum(NoncurrentVersionTransitionLifecycleRuleCount)/sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Noncurrent version expiration lifecycle rule count (Nombre de règles de cycle de vie d'expiration de version ancienne)	NoncurrentVersionExpirationLifecycleRuleCount	Nombre de règles de cycle de vie servant à faire expirer des versions d'objet anciennes	Avan	Optii ion des coût	N	-
Average noncurrent version expiration lifecycle rules per bucket (Nombre moyen de règles de cycle de vie d'expiration de version ancienne par compartiment)	-	Nombre moyen de règles de cycle de vie servant à faire expirer des versions d'objet anciennes	Avan	Optii ion des coût	Y	sum(NoncurrentVersionExpirationLifecycleRuleCount)/sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Abort incomplete multipart upload lifecycle rule count (Nombre de règles de cycle de vie d'abandon de chargements partitionnés non terminés)	AbortIncompleteMPULifecycleRuleCount	Nombre de règles de cycle de vie servant à supprimer les chargements partitionnés non terminés	Avan	Optin des coût	N	-
Average abort incomplete multipart upload lifecycle rules per bucket (Nombre moyen de règles de cycle de vie d'abandon de chargements partitionnés non terminés par compartiment)	-	Nombre moyen de règles de cycle de vie servant à supprimer les chargements partitionnés non terminés	Avan	Optin des coût	Y	sum(AbortIncompleteMPULifecycleRuleCount)/sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Expired object delete marker lifecycle rule count (Nombre de règles de cycle de vie des marqueurs de suppression d'objets expirés)	ExpiredObjectDeleteMarkerLifecycleRuleCount	Nombre de règles de cycle de vie pour supprimer les marqueurs de suppression des objets expirés	Avan	Optin des coût	N	-
Average expired object delete marker lifecycle rules per bucket (Nombre moyen de règles de cycle de vie des marqueurs de suppression d'objets expirés par compartiment)	-	Nombre moyen de règles de cycle de vie servant à supprimer les marqueurs de suppression des objets expirés	Avan	Optin des coût	Y	sum(ExpiredObjectDeleteMarkerLifecycleRuleCount)/sum(DistinctNumberOfBuckets)
Total lifecycle rule count (Nombre total de règles de cycle de vie)	TotalLifecycleRuleCount	Nombre total de règles de cycle de vie	Avan	Optin des coût	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Average lifecycle rule count per bucket (Nombre moyen de règles de cycle de vie par compartiment)	-	Nombre moyen de règles de cycle de vie	Avan	Opti ion des coût	Y	sum(Total LifecycleRuleCount)/ sum(DistinctNumberOfBuckets)
Encrypted bytes (Octets chiffrés)	Encrypted StorageBytes	Nombre total d'octets chiffrés	Free	Prot n des donr	N	-
% encrypted bytes (% d'octets chiffrés)	-	Pourcentage du total des octets chiffrés	Free	Prot n des donr	Y	sum(EncryptedObjectCount)/ s um(StorageBytes)
Encrypted object count (Nombre d'objets chiffrés)	Encrypted ObjectCount	Nombre total d'objets chiffrés	Free	Prot n des donr	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% encrypted objects (% d'objets chiffrés)	-	Pourcentage des objets chiffrés	Free	Prot n des donr	Y	sum(Encry ptedStora geBytes)/ sum(Objec tCount)
Unencrypted bytes (Octets non chiffrés)	Unencrypt edStorage Bytes	Nombre d'octets qui ne sont pas chiffrés	Free	Prot n des donr	Y	sum(Stora geBytes) - sum(Encry ptedStora geBytes)
% unencrypt ed bytes (% d'octets non chiffrés)	-	Pourcentage d'octets qui ne sont pas chiffrés	Free	Prot n des donr	Y	sum(Unenc ryptedSto rageBytes)/ sum(Sto rageBytes)
Unencrypt ed object count (Nombre d'objets non chiffrés)	Unencrypt edObjectCount	Nombre total d'objets non chiffrés	Free	Prot n des donr	Y	sum(Objec tCount) - sum(Encry ptedObjec tCount)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% unencrypted objects (% d'objets non chiffrés)	-	Pourcentage d'objets non chiffrés	Free	Prot n des donr	Y	sum(UnencryptedStorageBytes)/ sum(ObjectCount)
Replicated storage bytes source (Source d'octets de stockage répliqués)	ReplicatedStorageBytesSource	Nombre total d'octets répliqués à partir du compartiment source	Free	Prot n des donr	N	-
% replicated bytes source (Source en % d'octets répliqués)	-	Pourcentage du nombre total d'octets répliqués à partir du compartiment source	Free	Prot n des donr	Y	sum(ReplicatedStorageBytesSource)/ sum(StorageBytes)
Replicated object count source (Source du nombre d'objets répliqués)	ReplicatedObjectCountSource	Nombre d'objets répliqués à partir du compartiment source	Free	Prot n des donr	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% replicated objects source (Source en % d'objets répliqués)	-	Pourcentage du nombre total d'objets répliqués à partir du compartiment source	Free	Prot n des donr	Y	sum(ReplicatedStorageObjectCount)/ sum(ObjectCount)
Replication storage bytes destination (Destination des octets de stockage pour la réplication)	ReplicatedStorageBytes	Nombre total d'octets répliqués vers le compartiment de destination	Free	Prot n des donr	Y	-
% replicated bytes destination (Destination en % d'octets répliqués)	-	Pourcentage du nombre total d'octets répliqués vers le compartiment de destination	Free	Prot n des donr	Y	sum(ReplicatedStorageBytes) / sum(StorageBytes)
Replicated object count destination (Destination du nombre d'objets répliqués)	ReplicatedObjectCount	Nombre d'objets répliqués vers le compartiment de destination	Free	Prot n des donr	Y	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
% replicated objects destination (Destination en % des objets répliqués)	-	Pourcentage du nombre total d'objets répliqués vers le compartiment de destination	Free	Prot n des donr	Y	sum(ReplicatedObjectCount)/ sum(ObjectCount)
Object Lock bytes (Octets de verrouillage d'objets)	ObjectLockEnabledStorageBytes	Nombre total d'octets de stockage avec verrouillage d'objets activé	Free	Prot n des donr	Y	sum(UnencryptedStorageBytes)/ sum(ObjectLockEnabledStorageCount) - sum(ObjectLockEnabledStorageBytes)
% Object Lock bytes (% d'octets de verrouillage d'objets)	-	Pourcentage d'octets de stockage avec verrouillage d'objets activé	Free	Prot n des donr	Y	sum(ObjectLockEnabledStorageBytes)/ sum(StorageBytes)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Object Lock object count (Nombre d'objets de verrouillage d'objets)	ObjectLockEnabledObjectCount	Nombre total d'objets de verrouillage d'objets	Free	Prot n des donr	Y	-
% Object Lock objects (% d'objets avec verrouillage d'objets)	-	Pourcentage du nombre total d'objets pour lesquels le verrouillage d'objets est activé	Free	Prot n des donr	Y	sum(ObjectLockEnabledObjectCount)/ sum(ObjectCount)
Versioning-enabled bucket count (Nombre de compartiments activés pour la gestion des versions)	VersioningEnabledBucketCount	Nombre de compartiments pour lesquels la gestion des versions S3 est activée	Free	Prot n des donr	N	-
% versioning-enabled buckets (% de compartiments activés pour la gestion des versions)	-	Pourcentage de compartiments pour lesquels la gestion des versions S3 est activée	Free	Prot n des donr	Y	sum(VersioningEnabledBucketCount)/ sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
MFA delete-enabled bucket count (Nombre de compartiments activés pour la suppression MFA)	MFADeleteEnabledBucketCount	Nombre de compartiments pour lesquels la suppression MFA (authentification multifactorielle) est activée	Free	Prot n des donr	N	-
% MFA delete-enabled buckets (% de compartiments avec suppression MFA activée)	-	Pourcentage de compartiments pour lesquels la suppression MFA (authentification multifactorielle) est activée	Free	Prot n des donr	Y	sum(MFADeleteEnabledBucketCount)/sum(DistinctNumberOfBuckets)
SSE-KMS enabled bucket count (Nombre de compartiments avec SSE-KMS activé)	SSEKMSEnabledBucketCount	Nombre de compartiments qui utilisent le chiffrement côté serveur avec des clés AWS Key Management Service (SSE-KMS) pour le chiffrement de compartiment par défaut	Free	Prot n des donr	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% SSE-KMS enabled buckets (% de compartiments avec SSE-KMS activé)	-	Pourcentage de compartiments utilisant le protocole SSE-KMS pour le chiffrement des compartiments par défaut	Free	Prot n des donr	Y	sum(SSEKMSEnabledBucketCount)/ sum(DistinctNumberOfBuckets)
All unsupported signature requests (Toutes les demandes de signature non prises en charge)	AllUnsupportedSignatureRequests	Nombre total de demandes qui utilisent des versions de signature AWS non prises en charge	Avan	Prot n des donr	N	-
% all unsupported signature requests (% de toutes les demandes de signature non prises en charge)	-	Pourcentage de demandes qui utilisent des versions de signature AWS non prises en charge	Avan	Prot n des donr	Y	sum(AllUnsupportedSignatureRequests) / sum(AllRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
All unsupported TLS requests (Toutes les demandes TLS non prises en charge)	AllUnsupportedTLSRequests	Nombre de demandes qui utilisent des versions de protocole TLS (Transport Layer Security) non prises en charge	Avan	Prot n des donr	N	-
% all unsupported TLS requests (% de toutes les demandes TLS non prises en charge)	-	Pourcentage de demandes qui utilisent des versions TLS non prises en charge	Avan	Prot n des donr	Y	sum(AllUnsupportedTLSRequests)/sum(AllRequests)
All SSE-KMS requests (Toutes les demandes SSE-KMS)	AllSSEKMSRequests	Nombre total de demandes spécifiant le protocole SSE-KMS	Avan	Prot n des donr	N	-
% all SSE-KMS requests (% de toutes les demandes SSE-KMS)	-	Pourcentage de demandes qui spécifient le protocole SSE-KMS	Avan	Prot n des donr	Y	sum(AllSSEKMSRequests)/sum(AllRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Same-Region Replication rule count (Nombre de règles de réplication pour la même région)	SameRegionReplicationRuleCount	Nombre de règles de la réplication pour la réplication dans une même région (SRR)	Avan	Prot n des donr	N	-
Average Same-Region Replication rules per bucket (Nombre moyen de règles de réplication pour la même région par compartiment)	-	Nombre moyen de règles de réplication pour la réplication dans une même région (SRR)	Avan	Prot n des donr	Y	sum(SameRegionReplicationRuleCount)/sum(DistinctNumberOfBuckets)
Cross-Region Replication rule count (Nombre de règles de réplication entre régions)	CrossRegionReplicationRuleCount	Nombre de règles de réplication pour la réplication entre régions (CRR)	Avan	Prot n des donr	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Average Cross-Region Replication rules per bucket (Nombre moyen de règles de réplication entre régions par compartiment)	-	Nombre moyen de règles de réplication pour la réplication entre régions (CRR)	Avan	Prot n des donr	Y	sum(CrossRegionReplicationRuleCount) / sum(DistinctNumberOfBuckets)
Same-account replication rule count (Nombre de règles de réplication pour le même compte)	SameAccountReplicationRuleCount	Nombre de règles de réplication pour la réplication au sein du même compte	Avan	Prot n des donr	N	-
Average same-account replication rules per bucket (Nombre moyen de règles de réplication pour le même compte par compartiment)	-	Nombre moyen de règles de réplication pour la réplication au sein du même compte	Avan	Prot n des donr	Y	sum(SameAccountReplicationRuleCount) / sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Cross-account replication rule count (Nombre de règles de réplication entre comptes)	CrossAccountReplicationRuleCount	Nombre de règles de réplication pour la réplication entre comptes	Avan	Prot n des donr	N	-
Average cross-account replication rules per bucket (Nombre moyen de règles de réplication entre comptes par compartiment)	-	Nombre moyen de règles de réplication pour la réplication entre comptes	Avan	Prot n des donr	Y	sum(CrossAccountReplicationRuleCount)/sum(DistinctNumberOfBuckets)
Invalid destination replication rule count (Nombre de règles de réplication de destination non valide)	InvalidDestinationReplicationRuleCount	Nombre de règles de réplication dont la destination de réplication n'est pas valide	Avan	Prot n des donr	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
Average invalid destination replication rules per bucket (Nombre moyen de règles de réplication de destination non valide par compartiment)	-	Nombre moyen de règles de réplication dont la destination de réplication n'est pas valide	Avan	Prot n des donr	Y	sum(InvalidReplicationRuleCount)/sum(DistinctNumberOfBuckets)
Total replication rule count (Nombre total de règles de réplication)	-	Nombre total de règles de réplication	Avan	Prot n des donr	Y	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Average replication rule count per bucket (Nombre moyen de règles de réplication par compartiment)	-	Nombre total moyen de règles de réplication	Avan	Prot n des donr	Y	sum(toute s les métriques de nombres de règles de réplicati on)/ sum(D istinctNu mberOfBuc kets)
Object Ownership bucket owner enforced bucket count (Nombre de compartiments avec propriété d'objets Appliqué par le propriétaire du compartiment)	ObjectOwn ershipBuc ketOwnerE nforcedBu cketCount	Nombre total de compartiments pour lesquels les listes de contrôle d'accès (ACL) sont désactivé es en utilisant le paramètre Appliqué par le propriétaire du compartiment pour la propriété d'objets	Free	Gest de l'acc	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% Object Ownership bucket owner enforced buckets (% de compartiments avec propriété d'objets Appliqué par le propriétaire du compartiment)	-	Pourcentage de compartiments pour lesquels les listes ACL sont désactivé es en utilisant le paramètre Appliqué par le propriétaire du compartiment pour la propriété d'objets	Free	Gest de l'acc	Y	sum(Objec tOwnershi pBucketOw nerEnforc edBucketC ount)/ sum (Distinct NumberOfB uckets)
Object Ownership bucket owner preferred bucket count (Nombre de compartiments avec propriété d'objets Préféré par le propriétaire du compartiment)	ObjectOwn ershipBuc ketOwnerP referredB ucketCount	Nombre total de compartiments qui utilisent le paramètre Préféré par le propriétaire du compartiment pour la propriété d'objets	Free	Gest de l'acc	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
% Object Ownership bucket owner preferred buckets (% de compartiments avec propriété d'objets Préféré par le propriétaire du compartiment)	-	Pourcentage de compartiments qui utilisent le paramètre Préféré par le propriétaire du compartiment pour la propriété d'objets	Free	Gest de l'acc	Y	sum(ObjectOwnershipBucketOwnerPreferredBucketCount)/ sum(DistinctNumberOfBuckets)
Object Ownership object writer bucket count (Nombre de compartiments de l'enregistreur d'objets de la propriété d'objets)	ObjectOwnershipObjectWriterBucketCount	Nombre total de compartiments qui utilisent le paramètre d'enregistreur d'objets pour la propriété d'objets	Free	Gest de l'acc	N	-
% Object Ownership object writer buckets (% de compartiments de l'enregistreur d'objets de la propriété d'objets)	-	Pourcentage de compartiments qui utilisent le paramètre d'enregistreur d'objets pour la propriété d'objets	Free	Gest de l'acc	Y	sum(ObjectOwnershipObjectWriterBucketCount)/ sum(DistinctNumberOfBuckets)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Transfer Accelerat ion enabled bucket count (Nombre de compartiments activés pour l'accélération de transfert)	TransferA ccelerati onEnabled BucketCount	Nombre total de compartiments pour lesquels l'accélération de transfert est activée	Free	Perf ces	N	-
% Transfer Accelerat ion enabled buckets (Pourcentage de compartim ents activés pour l'accélér ation de transfert)	-	Pourcentage de compartiments pour lesquels l'accélération de transfert est activée	Free	Perf ces	Y	sum(Trans ferAccele rationEna bledBucke tCount)/ s um(Distin ctNumberO fBuckets)
Event Notificat ion enabled bucket count (Nombre de compartim ents activés par notification d'événements)	EventNoti ficationE nabledBuc ketCount	Nombre total de compartiments pour lesquels les notifications d'événements sont activées	Free	Évér s	N	

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
% Event Notification enabled buckets (% de compartiments activés par notification d'événements)	-	Pourcentage de compartiments pour lesquels les notifications d'événements sont activées	Free	Évér s	Y	sum(EventNotificationEnabledBucketCount)/sum(DistinctNumberOfBuckets)
Toutes les demandes	AllRequests	Nombre total de requêtes effectuées	Avan	Activ	N	-
Demandes Get	GetRequests	Nombre total de requêtes `GET` effectuées	Avan	Activ	N	-
Put Requests (Requêtes PUT)	PutRequests	Nombre total de requêtes `PUT` effectuées	Avan	Activ	N	-
Head Requests (Requêtes HEAD)	HeadRequests	Nombre total de requêtes `HEAD` effectuées	Avan	Activ	N	-
Delete Requests (Requêtes DELETE)	DeleteRequests	Nombre total de requêtes `DELETE` effectuées	Avan	Activ	N	-

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Demandes List	ListRequests	Nombre total de requêtes `LIST` effectuées	Avan	Activ	N	-
Post Requests (Requêtes POST)	PostRequests	Nombre total de requêtes `POST` effectuées	Avan	Activ	N	-
Select Requests (Requêtes SELECT)	SelectRequests	Nombre total de requêtes S3 SELECT	Avan	Activ	N	-
Select scanned bytes (Octets de sélection analysés)	SelectScannedBytes	Nombre d'octets de sélection S3 analysés	Avan	Activ	N	-
Select returned bytes (Octets de sélection retournés)	SelectReturnedBytes	Nombre d'octets de sélection S3 retournés	Avan	Activ	N	-
Octets téléchargés	BytesDownloaded	Nombre d'octets téléchargés	Avan	Activ	N	-
% retrieval rate (Taux d'extraction en %)	-	Pourcentage d'octets téléchargés	Avan	Activ	Y	sum(BytesDownloaded)/ sum(StorageBytes)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Octets chargés	BytesUploaded	Nombre d'octets chargés	Avan	Activ	N	-
% ingest ratio (Ratio d'ingestion en %)	-	Pourcentage d'octets chargés	Avan	Activ	Y	sum(BytesUploaded) / sum(StorageBytes)
4xx errors (Erreurs 4xx)	4xxErrors	Nombre total de codes de statut HTTP 4xx	Avan	Activ	N	-
5xx errors (Erreurs 5xx)	5xxErrors	Nombre total de codes de statut HTTP 5xx	Avan	Activ	N	-
Total errors (Nombre total d'erreurs)	-	Somme de toutes les erreurs 4xx et 5xx	Avan	Activ	Y	sum(4xxErrors) + sum(5xxErrors)
% error rate (Taux d'erreurs en %)	-	Nombre total d'erreurs 4xx et 5xx exprimé en pourcentage du nombre total de demandes	Avan	Activ	Y	sum(TotalErrors)/ sum(TotalRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
Nombre de codes de statut 200 OK	200OKStatusCount	Nombre total de codes de statut 200 OK	Avan	Cod de statu déta	N	-
% 200 OK status (% de codes de statut 200 OK)	-	Nombre total de codes de statut 200 OK sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(200OKStatusCount)/sum(AllRequests)
206 Partial Content status count (Nombre de codes de statut 206 Contenu partiel)	206PartialContentStatusCount	Nombre total de codes de statut 206 Contenu partiel	Avan	Cod de statu déta	N	-
% 206 Partial Content status (% de codes de statut 206 Contenu partiel)	-	Nombre total de codes de statut 206 Contenu partiel sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(206PartialContentStatusCount)/sum(AllRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
400 Bad Request error count (Nombre d'erreurs 400 Requête erronée)	400BadRequestErrorCount	Nombre total de codes de statut 400 Requête erronée	Avan	Cod de statu déta	N	-
% 400 Bad Request errors (% d'erreurs 400 Requête erronée)	-	Nombre total de codes de statut 400 Requête erronée sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(400BadRequestErrorCount)/ sum(All Requests)
403 Forbidden error count (Nombre d'erreurs 403 Interdit)	403ForbiddenErrorCount	Nombre total de codes de statut 403 Interdit	Avan	Cod de statu déta	N	-
% 403 Forbidden errors (% d'erreurs 403 Interdit)	-	Nombre total de codes de statut 403 Interdit sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(403ForbiddenErrorCount) / sum(AllRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métr dériv
404 Not Found error count (Nombre d'erreurs 404 Non trouvé)	404NotFoundErrorCount	Nombre total de codes de statut 404 Non trouvé	Avan	Cod de statu déta	N	-
% 404 Not Found errors (% d'erreurs 404 Non trouvé)	-	Nombre total de codes de statut 404 Non trouvé sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(404NotFoundErrorCount)/ sum(AllRequests)
500 Internal Server Error count (Nombre de codes 500 Erreur de serveur interne)	500InternalServerErrorCount	Nombre total de codes de statut 500 Erreur de serveur interne	Avan	Cod de statu déta	N	-
% 500 Internal Server Errors (% d'erreurs 500 Erreur de serveur interne)	-	Nombre total de codes de statut 500 Erreur de serveur interne sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(500InternalServerErrorCount)/ sum (AllRequests)

Nom de la métrique	CloudWatch et exportation	Description	Nive	Caté[2]	D	Form de métri dériv
503 Service Unavailable error count (Nombre d'erreurs 503 Service non disponible)	503ServiceUnavailableErrorCount	Nombre total de codes de statut 503 Service non disponible	Avan	Cod de statu déta	N	-
% 503 Service Unavailable errors (% d'erreurs 503 Service non disponible)	-	Nombre total de codes de statut 503 Service non disponible sous la forme d'un pourcentage du nombre total de demandes	Avan	Cod de statu déta	Y	sum(503ServiceUnavailableErrorCount) / sum(AllRequests)

[1] Toutes les métriques de stockage du niveau d'offre gratuite sont disponibles au niveau du groupe Storage Lens. Les métriques de niveau avancé ne sont pas disponibles au niveau du groupe Storage Lens.

[2] Les métriques relatives au nombre de règles et aux paramètres des compartiments ne sont pas disponibles au niveau du préfixe.

Utilisation d'Amazon S3 Storage Lens à l'aide de la console et de l'API

Amazon S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Vous pouvez utiliser les métriques S3 Storage Lens pour générer des informations récapitulatives, telles que la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes qui connaissent la croissance la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier les opportunités

d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection et de sécurisation des données et améliorer les performances des charges de travail d'application. Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour faire expirer les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

La section suivante contient des exemples de création, de mise à jour et d'affichage des configurations S3 Storage Lens et d'exécution d'opérations liées à la fonction. Si vous utilisez S3 Storage Lens avec AWS Organizations, ces exemples couvrent également ces cas d'utilisation. Dans les exemples, remplacez les valeurs de variable par celles qui vous conviennent.

Rubriques

Utilisation d'Amazon S3 Storage Lens dans la console

Amazon S3 Storage Lens est une fonction d'analyse du stockage dans le cloud que vous pouvez utiliser pour obtenir une visibilité à l'échelle de l'organisation sur l'utilisation et l'activité du stockage d'objets. Vous pouvez utiliser les métriques S3 Storage Lens pour générer des informations récapitulatives, telles que la quantité de stockage dont vous disposez dans l'ensemble de votre organisation ou les compartiments et préfixes qui connaissent la croissance la plus rapide. Vous pouvez également utiliser les métriques S3 Storage Lens pour identifier les opportunités

d'optimisation des coûts, mettre en œuvre les bonnes pratiques de protection et de sécurisation des données et améliorer les performances des charges de travail d'application. Par exemple, vous pouvez identifier les compartiments qui sont dépourvus de règles de cycle de vie S3 pour faire expirer les chargements partitionnés non terminés datant de plus de 7 jours. Vous pouvez également identifier les compartiments qui ne respectent pas les bonnes pratiques de protection des données, telles que l'utilisation de la réplication S3 ou de la gestion des versions S3. S3 Storage Lens analyse également les métriques de stockage pour fournir des recommandations contextuelles afin d'aider à réduire les coûts de stockage et à appliquer les bonnes pratiques de protection des données.

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

> **Note**
>
> L'affichage ou la visualisation des mises à jour apportées à la configuration de votre tableau de bord peut prendre jusqu'à 48 heures.

Rubriques

- Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens
- Désactiver ou supprimer des tableaux de bord Amazon S3 Storage Lens
- Utilisation AWS Organizations pour créer des tableaux de bord au niveau de l'organisation

Créer et mettre à jour les tableaux de bord Amazon S3 Storage Lens

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre

tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

Le tableau de bord par défaut d'Amazon S3 Storage Lens est default-account-dashboard. Ce tableau de bord est préconfiguré par Amazon S3 pour vous aider à visualiser le résumé des informations et des tendances des métriques gratuites et avancées agrégées de votre compte dans la console. Vous ne pouvez pas modifier la portée de la configuration du tableau de bord par défaut, mais vous pouvez mettre à niveau la sélection de métriques gratuites vers les métriques et recommandations avancées payantes, configurer l'exportation facultative des métriques ou même désactiver le tableau de bord par défaut. Le tableau de bord par défaut ne peut pas être supprimé.

Vous pouvez également créer des tableaux de bord personnalisés S3 Storage Lens supplémentaires qui peuvent être adaptés à votre organisation, à des régions AWS Organizations ou à des compartiments spécifiques au sein d'un compte.

Créer un tableau de bord Amazon S3 Storage Lens

Procédure pour créer un tableau de bord Amazon S3 Storage Lens sur la console Amazon S3.

Étape 1 : définir la portée du tableau de bord

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom de la AWS région actuellement affichée. Choisissez ensuite la région vers laquelle vous souhaitez passer.
3. Dans le panneau de navigation de gauche, sous S3 Storage Lens, choisissez Tableaux de bord.
4. Choisissez Create dashboard (Créer un tableau de bord).
5. Sur la page Tableau de bord, dans la section General (Général), procédez comme suit :

    a. Consultez la région d'origine de votre tableau de bord. La région d'origine est l' Région AWS endroit où sont stockées la configuration et les mesures de ce tableau de bord Storage Lens.

    b. Saisissez un nom de tableau de bord.

    Les noms de tableau de bord doivent contenir moins de 65 caractères et ne doivent pas contenir de caractères spéciaux ou d'espaces.

> **Note**
>
> Vous ne pouvez pas modifier ce nom de tableau de bord une fois le tableau de bord créé.

c. Vous pouvez éventuellement choisir d'ajouter des balises à votre tableau de bord. Vous pouvez utiliser des balises pour gérer les autorisations de votre tableau de bord et suivre les coûts relatifs à S3 Storage Lens.

   Pour plus d'informations, consultez la section Contrôle de l'accès à l'aide d'étiquettes de ressources du Guide de l'utilisateur IAM, et la section Étiquettes de répartition des coûts générées par AWS du Guide de l'utilisateur AWS Billing .

   > **Note**
   >
   > Vous pouvez ajouter jusqu'à 50 balises à votre configuration de tableau de bord.

6. Dans la section Dashboard scope (Portée du tableau de bord), procédez comme suit :

   a. Sélectionnez les Régions et les compartiments que vous souhaitez que S3 Storage Lens inclue ou exclue du tableau de bord.

   b. Sélectionnez les compartiments dans les Régions sélectionnées que vous souhaitez que S3 Storage Lens inclue ou exclue. Vous pouvez inclure ou exclure des compartiments, mais pas les deux. Cette option n'est pas disponible lorsque vous créez des tableaux de bord au niveau de l'organisation.

> **Note**
>
> - Vous pouvez inclure ou exclure des Régions et des compartiments. Cette option est limitée aux Régions uniquement lors de la création de tableaux de bord au niveau de l'organisation sur les comptes membres de votre organisation.
> - Vous pouvez choisir jusqu'à 50 compartiments à inclure ou exclure.

Étape 2 : configurer la sélection des métriques

1. Dans la section Metrics selection (Sélection de métriques), sélectionnez le type de métriques que vous souhaitez agréger pour ce tableau de bord.

   - Pour inclure des métriques gratuites agrégées au niveau du compartiment et disponibles pour les requêtes pendant 14 jours, choisissez Free Metrics (Métriques gratuites).
   - Pour activer les métriques avancées et d'autres options avancées, choisissez Advanced metrics and recommendations (Métriques et recommandations avancées). Ces options incluent l'agrégation avancée des préfixes, la CloudWatch publication sur Amazon et les recommandations contextuelles. Les données sont disponibles pour les requêtes pendant 15 mois. Les métriques et recommandations avancées entraînent un coût supplémentaire. Pour de plus amples informations, consultez la tarification Amazon S3.

     Pour plus d'informations sur les métriques gratuites et avancées, consultez Sélection des métriques.
2. Sous Advanced metrics and recommendations features (Fonctionnalités de métriques et recommandations avancées), sélectionnez les options que vous voulez activer :

   - Advanced metrics (Métriques avancées)
   - CloudWatch publication
   - Prefix aggregation (Agrégation de préfixes)

   > **Important**
   >
   > Si vous activez l'agrégation de préfixes pour votre configuration S3 Storage Lens, les métriques au niveau du préfixe ne seront pas publiées sur. CloudWatch Seules les métriques S3 Storage Lens au niveau du bucket, du compte et de l'organisation sont publiées sur. CloudWatch
3. Si vous avez activé Advanced metrics (Métriques avancées), sélectionnez les Advanced metrics categories (Catégories de métriques avancées) que vous souhaitez afficher dans votre tableau de bord S3 Storage Lens :

   - Métriques d'activité
   - Detailed status code metrics (Métriques détaillées sur le code de statut)
   - Advanced cost optimization metrics (Métriques avancées sur l'optimisation des coûts)

- Advanced data protection metrics (Métriques avancées sur la protection des données)

Pour plus d'informations sur les catégories de métriques, consultez Catégories de métriques. Pour obtenir une liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.

4. Si vous avez choisi d'activer l'agrégation des préfixes, configurez les éléments suivants :

   a. Choisissez la taille minimale du seuil de préfixe pour ce tableau de bord.

      Par exemple, un seuil de préfixe de 5 % indique que les préfixes représentant 5 % ou plus de la taille totale de stockage du compartiment seront agrégés.

   b. Sélectionnez la profondeur de préfixe.

      Ce paramètre indique le nombre maximal de niveaux jusqu'auquel les préfixes sont évalués. La profondeur de préfixe doit être inférieure à 10.

   c. Saisissez un caractère de délimiteur de préfixe.

      Cette valeur est utilisée pour identifier chaque niveau de préfixe. La valeur par défaut dans Amazon S3 est le caractère `/`, mais votre structure de stockage peut utiliser d'autres délimiteurs.

(Facultatif) Étape 3 : exporter des métriques pour le tableau de bord

1. Dans la section Metrics export (Exportation de métriques), choisissez Enable (Activer) pour créer une exportation de métriques qui sera placée quotidiennement dans un compartiment de destination de votre choix.

   L'exportation des métriques est au format CSV ou Apache Parquet. La portée des données est la même que celle des données de votre tableau de bord S3 Storage Lens sans les recommandations.

2. Si vous avez activé l'exportation de métriques, choisissez le format de sortie de votre exportation quotidienne de métriques : CSV ou Apache Parquet.

   Parquet est un format de fichier open source pour Hadoop, qui stocke les données imbriquées dans un format en colonnes plat.

3. Sélectionnez le compartiment S3 de destination pour l'exportation de vos métriques.

Vous pouvez choisir un compartiment dans le compte actuel du tableau de bord S3 Storage Lens. Vous pouvez également en choisir un autre Compte AWS si vous disposez des autorisations du compartiment de destination et de l'ID de compte du propriétaire du compartiment de destination.

4. Choisissez le compartiment S3 de destination (format : `s3://bucket-name/prefix`).

   Le compartiment doit être dans la région d'origine de votre tableau de bord S3 Storage Lens. La console S3 vous montre le paramètre Destination bucket permission (Autorisation de compartiment de destination) qui sera ajouté par Amazon S3 à la politique de compartiment de destination. Amazon S3 met à jour la politique de compartiment sur le compartiment de destination pour permettre à S3 de placer des données dans ce compartiment.

5. (Facultatif) Pour activer le chiffrement côté serveur pour l'exportation de vos métriques, choisissez Specify an encryption key (Spécifier une clé de chiffrement). Ensuite, choisissez le Type de chiffrement : Clés gérées par Amazon S3 (SSE-S3) ou Clé AWS Key Management Service (SSE-KMS).

   Vous pouvez choisir entre une clé gérée par Amazon S3 (SSE-S3) et une clé AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Facultatif) Pour spécifier une AWS KMS clé, vous devez choisir une clé KMS ou saisir une clé Amazon Resource Name (ARN).

   Si vous choisissez une clé gérée par le client, vous devez accorder à S3 Storage Lens l'autorisation de chiffrer dans la politique de clé AWS KMS . Pour plus d'informations, consultez Utilisation d'un AWS KMS key pour chiffrer vos exportations de métriques.

7. Choisissez Create dashboard (Créer un tableau de bord).

Pour améliorer la visibilité de votre stockage, vous pouvez créer un ou plusieurs groupes S3 Storage Lens et les attacher à votre tableau de bord. Un groupe S3 Storage Lens est un filtre défini personnalisé pour les objets basé sur les préfixes, les suffixes, les balises d'objet, la taille d'objet, l'âge d'objet ou une combinaison de ces filtres.

Vous pouvez utiliser les groupes S3 Storage Lens pour obtenir une visibilité précise sur les compartiments partagés de grande taille, tels que les lacs de données, afin de prendre des décisions commerciales plus éclairées. Par exemple, vous pouvez rationaliser l'allocation du stockage et optimiser les rapports sur les coûts en répartissant l'utilisation du stockage en groupes d'objets

spécifiques pour des projets individuels et des centres de coûts au sein d'un compartiment ou de plusieurs compartiments.

Pour utiliser les groupes S3 Storage Lens, vous devez mettre à niveau votre tableau de bord afin d'utiliser des métriques et des recommandations avancées. Pour plus d'informations sur les groupes S3 Storage Lens, consultez the section called “Utilisation des groupes S3 Storage Lens”.

Mettre à jour un tableau de bord Amazon S3 Storage Lens

Suivez les étapes suivantes pour mettre à jour un tableau de bord Amazon S3 Storage Lens sur la console Amazon S3.

Étape 1 : mettre à jour la portée du tableau de bord

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Storage Lens, Tableaux de bord).
3. Choisissez le tableau de bord que vous souhaitez modifier, puis choisissez Edit (Modifier).

   La page Edit dashboard (Modifier le tableau de bord) s'ouvre.

   > **Note**
   >
   > Vous ne pouvez pas modifier les éléments suivants :
   >
   > - Le nom du tableau de bord
   > - La Région d'accueil
   > - La portée de tableau de bord du tableau de bord par défaut, qui est étendue à l'ensemble du stockage de votre compte.

4. (Facultatif) Sur la page de configuration du tableau de bord, dans la section General (Général), mettez à jour et ajoutez des étiquettes dans votre tableau de bord.

   Vous pouvez utiliser des balises pour gérer les autorisations de votre tableau de bord et pour suivre les coûts liés à S3 Storage Lens. Pour plus d'informations, consultez la section Contrôle de l'accès à l'aide d'étiquettes de ressources du Guide de l'utilisateur IAM, et la section Étiquettes de répartition des coûts générées par AWS du Guide de l'utilisateur AWS Billing .

> **Note**
>
> Vous pouvez ajouter jusqu'à 50 balises à votre configuration de tableau de bord.

5. Dans la section Dashboard scope (Portée du tableau de bord), procédez comme suit :

   a. Mettez à jour les Régions et les compartiments que S3 Storage Lens doit inclure ou exclure du tableau de bord.

   > **Note**
   >
   > - Vous pouvez inclure ou exclure des Régions et des compartiments. Cette option est limitée aux Régions uniquement lors de la création de tableaux de bord au niveau de l'organisation sur les comptes membres de votre organisation.
   > - Vous pouvez choisir jusqu'à 50 compartiments à inclure ou exclure.

   b. Mettez à jour les compartiments des Régions sélectionnées que vous souhaitez inclure ou exclure de S3 Storage Lens. Vous pouvez inclure ou exclure des compartiments, mais pas les deux. Cette option n'est pas disponible lors de la création de tableaux de bord au niveau de l'organisation.

Étape 2 : mettre à jour la sélection de métriques

1. Dans la section Metrics selection (Sélection de métriques), sélectionnez le type de métriques que vous souhaitez agréger pour ce tableau de bord.

   - Pour inclure des métriques gratuites agrégées au niveau du compartiment et disponibles pour les requêtes pendant 14 jours, choisissez Free Metrics (Métriques gratuites).
   - Pour activer les métriques avancées et d'autres options avancées, choisissez Advanced metrics and recommendations (Métriques et recommandations avancées). Ces options incluent l'agrégation avancée des préfixes, la CloudWatch publication sur Amazon et les recommandations contextuelles. Les données sont disponibles pour les requêtes pendant 15 mois. Les métriques et recommandations avancées entraînent un coût supplémentaire. Pour de plus amples informations, consultez la tarification Amazon S3.

     Pour plus d'informations sur les métriques gratuites et avancées, consultez Sélection des métriques.

2. Sous Advanced metrics and recommendations features (Fonctionnalités de métriques et recommandations avancées), sélectionnez les options que vous voulez activer :

   - Advanced metrics (Métriques avancées)
   - CloudWatch publication
   - Prefix aggregation (Agrégation de préfixes)

   > **Important**
   >
   > Si vous activez l'agrégation de préfixes pour votre configuration S3 Storage Lens, les métriques au niveau du préfixe ne seront pas publiées sur. CloudWatch Seules les métriques S3 Storage Lens au niveau du bucket, du compte et de l'organisation sont publiées sur. CloudWatch

3. Si vous avez activé Advanced metrics (Métriques avancées), sélectionnez les Advanced metrics categories (Catégories de métriques avancées) que vous souhaitez afficher dans votre tableau de bord S3 Storage Lens :

   - Métriques d'activité
   - Detailed status code metrics (Métriques détaillées sur le code de statut)
   - Advanced cost optimization metrics (Métriques avancées sur l'optimisation des coûts)
   - Advanced data protection metrics (Métriques avancées sur la protection des données)

   Pour plus d'informations sur les catégories de métriques, consultez Catégories de métriques. Pour obtenir une liste complète des métriques, consultez Glossaire des métriques Amazon S3 Storage Lens.

4. Si vous avez choisi d'activer l'agrégation des préfixes, configurez les éléments suivants :

   a. Choisissez la taille minimale du seuil de préfixe pour ce tableau de bord.

      Par exemple, un seuil de préfixe de 5 % indique que les préfixes représentant 5 % ou plus de la taille totale de stockage du compartiment seront agrégés.

   b. Sélectionnez la profondeur de préfixe.

      Ce paramètre indique le nombre maximal de niveaux jusqu'auquel les préfixes sont évalués. La profondeur de préfixe doit être inférieure à 10.

c. Saisissez un caractère de délimiteur de préfixe.

Il s'agit de la valeur utilisée pour identifier chaque niveau de préfixe. La valeur par défaut dans Amazon S3 est le caractère `/`, mais votre structure de stockage peut utiliser d'autres délimiteurs.

(Facultatif) Étape 3 : exporter des métriques pour le tableau de bord

1. Dans la section Metrics export (Exportation de métriques), choisissez Enable (Activer) pour créer une exportation de métriques qui sera placée quotidiennement dans un compartiment de destination de votre choix. Pour désactiver l'exportation des métriques, choisissez Disable (Désactiver).

   L'exportation des métriques est au format CSV ou Apache Parquet. La portée des données est la même que celle des données de votre tableau de bord S3 Storage Lens sans les recommandations.
2. Si cette option est activée, choisissez le format de sortie de votre exportation quotidienne de métriques : CSV ou Apache Parquet.

   Parquet est un format de fichier open source pour Hadoop, qui stocke les données imbriquées dans un format en colonnes plat.
3. Sélectionnez le compartiment S3 de destination pour l'exportation de vos métriques.

   Vous pouvez choisir un compartiment dans le compte actuel du tableau de bord S3 Storage Lens. Vous pouvez également en choisir un autre Compte AWS si vous disposez des autorisations du compartiment de destination et de l'ID de compte du propriétaire du compartiment de destination.
4. Choisissez le compartiment S3 de destination (format : `s3://`*`bucket-name`*`/`*`prefix`*).

   Le compartiment doit être dans la région d'origine de votre tableau de bord S3 Storage Lens. La console S3 vous montre le paramètre Destination bucket permission (Autorisation de compartiment de destination) qui sera ajouté par Amazon S3 à la politique de compartiment de destination. Amazon S3 met à jour la politique de compartiment sur le compartiment de destination pour permettre à S3 de placer des données dans ce compartiment.
5. (Facultatif) Pour activer le chiffrement côté serveur pour l'exportation de vos métriques, choisissez Specify an encryption key (Spécifier une clé de chiffrement). Ensuite, choisissez

le Type de chiffrement : Clés gérées par Amazon S3 (SSE-S3) ou Clé AWS Key Management Service (SSE-KMS).

Vous pouvez choisir entre une clé gérée par Amazon S3 (SSE-S3) et une clé AWS Key Management Service (AWS KMS) (SSE-KMS).

6. (Facultatif) Pour spécifier une AWS KMS clé, vous devez choisir une clé KMS ou saisir une clé Amazon Resource Name (ARN). Sous CléAWS KMS , spécifiez votre clé KMS de l'une des manières suivantes :

   - Pour choisir parmi une liste de clés KMS disponibles, choisissez Choisir parmi vos clés AWS KMS keys, puis sélectionnez votre Clé KMS dans la liste des clés disponibles.

     La clé Clé gérée par AWS (`aws/s3`) et la clé gérée par votre client apparaissent toutes deux dans cette liste. Pour plus d'informations sur les clés gérées par le client, consultez Clés de client et clés AWS dans le Guide du développeur AWS Key Management Service .

     > **Note**
     >
     > Le Clé gérée par AWS (`aws/S3`) n'est pas pris en charge pour le chiffrement SSE-KMS avec S3 Storage Lens.

   - Pour saisir l'ARN de la clé KMS, choisissez Saisir l'ARN de AWS KMS key , puis saisissez l'ARN de votre clé KMS dans le champ qui s'affiche.
   - Pour créer une nouvelle clé gérée par le client dans la AWS KMS console, choisissez Create a KMS key.

     Si vous choisissez une clé gérée par le client, vous devez accorder à S3 Storage Lens l'autorisation de chiffrer dans la politique de clé AWS KMS . Pour plus d'informations, consultez Utilisation d'un AWS KMS key pour chiffrer vos exportations de métriques.

     Pour plus d'informations sur la création d'un AWS KMS key, consultez la section Création de clés dans le guide du AWS Key Management Service développeur.

7. Sélectionnez Enregistrer les modifications.

Pour améliorer la visibilité de votre stockage, vous pouvez créer un ou plusieurs groupes S3 Storage Lens et les attacher à votre tableau de bord. Un groupe S3 Storage Lens est un filtre défini personnalisé pour les objets basé sur les préfixes, les suffixes, les balises d'objet, la taille d'objet, l'âge d'objet ou une combinaison de ces filtres.

Vous pouvez utiliser les groupes S3 Storage Lens pour obtenir une visibilité précise sur les compartiments partagés de grande taille, tels que les lacs de données, afin de prendre des décisions commerciales plus éclairées. Par exemple, vous pouvez rationaliser l'allocation du stockage et optimiser les rapports sur les coûts en répartissant l'utilisation du stockage en groupes d'objets spécifiques pour des projets individuels et des centres de coûts au sein d'un compartiment ou de plusieurs compartiments.

Pour utiliser les groupes S3 Storage Lens, vous devez mettre à niveau votre tableau de bord afin d'utiliser des métriques et des recommandations avancées. Pour plus d'informations sur les groupes S3 Storage Lens, consultez the section called “Utilisation des groupes S3 Storage Lens”.

Désactiver ou supprimer des tableaux de bord Amazon S3 Storage Lens

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

Le tableau de bord par défaut d'Amazon S3 Storage Lens est default-account-dashboard. Ce tableau de bord est préconfiguré par Amazon S3 pour vous aider à visualiser le résumé des informations et des tendances des métriques gratuites et avancées agrégées de votre compte dans la console. Vous ne pouvez pas modifier la portée de la configuration du tableau de bord par défaut, mais vous pouvez mettre à niveau la sélection de métriques gratuites vers les métriques et recommandations avancées payantes, configurer l'exportation facultative des métriques ou même désactiver le tableau de bord par défaut. Le tableau de bord par défaut ne peut pas être supprimé.

Vous pouvez supprimer ou désactiver un tableau de bord Amazon S3 Storage Lens à partir de la console Amazon S3. Lorsqu'un tableau de bord est désactivé ou supprimé, il ne peut plus générer de métriques. Un tableau de bord désactivé conserve toujours ses informations de configuration. Ainsi, vous pouvez facilement le reprendre lorsque vous le réactivez. Un tableau de bord désactivé retient ses données historiques jusqu'à ce qu'il ne soit plus disponible pour les requêtes.

Les données des sélections de métriques gratuites sont disponibles pour les requêtes pendant 14 jours, et les données des sélections de métriques et recommandations avancées sont disponibles pour les requêtes pendant 15 mois.

Désactiver un tableau de bord Amazon S3 Storage Lens

Pour désactiver un tableau de bord S3 Storage Lens

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux), sélectionnez le tableau de bord que vous souhaitez désactiver, puis sélectionnez Disable (Désactiver) en haut de la liste.
4. Sur la page de confirmation, confirmez que vous souhaitez désactiver le tableau de bord en saisissant son nom dans le champ de texte, puis sélectionnez Confirmer.

Supprimer un tableau de bord Amazon S3 Storage Lens

> **Note**
>
> Vous ne pouvez pas supprimer le tableau de bord par défaut. Cependant, vous pouvez la ou le désactiver. Avant de supprimer un tableau de bord que vous avez créé, tenez compte des points suivants :
>
> - Au lieu de supprimer un tableau de bord, vous pouvez le désactiver afin qu'il puisse être réactivé à l'avenir. Pour plus d'informations, consultez Désactiver un tableau de bord Amazon S3 Storage Lens.
> - Lorsque le tableau de bord est supprimé, tous les paramètres de configuration qui lui sont associés sont également supprimés.
> - En outre, toutes les données historiques des métriques ne sont plus disponibles. Ces données historiques sont conservées pendant 15 mois. Si vous souhaitez accéder à nouveau à ces données, vous devez créer un tableau de bord portant le même nom dans la même Région d'accueil que celui qui a été supprimé.

Pour supprimer un tableau de bord S3 Storage Lens

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, Dashboards (Tableaux de bord).
3. Dans la liste Dashboards (Tableaux de bord), sélectionnez le tableau de bord à supprimer, puis Delete (Supprimer) en haut de la liste.
4. Sur la page Supprimer les tableaux de bord, confirmez que vous souhaitez supprimer le tableau de bord en saisissant son nom dans le champ de texte. Ensuite, choisissez Valider.

Utilisation AWS Organizations pour créer des tableaux de bord au niveau de l'organisation

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3.

Le tableau de bord par défaut d'Amazon S3 Storage Lens est default-account-dashboard. Ce tableau de bord est préconfiguré par Amazon S3 pour vous aider à visualiser le résumé des informations et des tendances des métriques gratuites et avancées agrégées de votre compte dans la console. Vous ne pouvez pas modifier la portée de la configuration du tableau de bord par défaut, mais vous pouvez mettre à niveau la sélection de métriques gratuites vers les métriques et recommandations avancées payantes, configurer l'exportation facultative des métriques ou même désactiver le tableau de bord par défaut. Le tableau de bord par défaut ne peut pas être supprimé.

Vous pouvez également créer des tableaux de bord S3 Storage Lens supplémentaires axés sur des compartiments S3 spécifiques Régions AWS ou autres Comptes AWS au sein de votre organisation.

Un tableau de bord S3 Storage Lens fournit une ressource précieuse d'informations sur sa portée de stockage. Un tableau de bord permet de visualiser plus de 30 métriques qui représentent les tendances et d'autres informations, notamment le résumé du stockage, la rentabilité, la protection des données et l'activité.

Amazon S3 Storage Lens peut être utilisé pour collecter des métriques de stockage et des données d'utilisation pour tous les comptes qui font partie de votre AWS Organizations hiérarchie. Pour ce faire, vous devez utiliser AWS Organizations et activer l'accès sécurisé à S3 Storage Lens à l'aide de votre compte AWS Organizations de gestion.

Une fois l'accès sécurisé activé, vous pouvez ajouter un accès administrateur délégué aux comptes de votre organisation. Ces comptes peuvent ensuite créer des tableaux de bord et des configurations à l'échelle de l'organisation pour S3 Storage Lens. Pour de plus amples informations sur l'activation de l'accès sécurisé, veuillez consulter la section Amazon S3 Lens et AWS Organizations du Guide de l'utilisateur AWS Organizations .

Les commandes de console suivantes ne sont disponibles que pour les comptes AWS Organizations de gestion.

Activer l'accès sécurisé de S3 Storage Lens à votre organisation

L'activation de l'accès sécurisé permet à Amazon S3 Storage Lens d'accéder à votre AWS Organizations hiérarchie, à vos membres et à votre structure par le biais AWS Organizations d'opérations d'API. S3 Storage Lens devient un service sécurisé pour l'ensemble de la structure de votre organisation. Il peut créer des rôles liés à un service dans les comptes de gestion ou d'administrateur délégué de votre organisation chaque fois qu'une configuration de tableau de bord est créée.

Le rôle lié à un service accorde des autorisations S3 Storage Lens pour décrire les organisations, répertorier les comptes, vérifier la liste des accès aux services pour les organisations et obtenir des administrateurs délégués pour l'organisation. Cela permet à S3 Storage Lens de collecter des métriques d'utilisation et d'activité du stockage intercomptes pour les tableaux de bord au sein des comptes de votre organisation.

Pour plus d’informations, consultez Utilisation des rôles liés à un service pour le cadre de stockage Amazon S3.

> **Note**
>
> - L'accès sécurisé ne peut être activé que par le compte de gestion.
> - Seuls le compte de gestion et les administrateurs délégués peuvent créer des tableaux de bord ou des configurations S3 Storage Lens pour votre organisation.

Pour permettre à S3 Storage Lens d'avoir un accès sécurisé

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, puis Organization settings (Paramètres de l'organisation).
3. Dans Accès aux organisations, sélectionnez Modifier.

   La page Accès aux organisations s'ouvre. Ici, vous pouvez sélectionner Enable trusted access (Activer l'accès sécurisé) pour S3 Storage Lens. Cela vous permet à vous, ainsi qu'à tout autre titulaire de compte que vous ajoutez en tant qu'administrateur délégué, de créer des tableaux de bord pour tous les comptes et le stockage de votre organisation.

Désactiver l'accès sécurisé de S3 Storage Lens à votre organisation

En désactivant l'accès sécurisé de S3 Storage Lens, vous limitez son fonctionnement au niveau du compte uniquement. Chacun des titulaires de compte ne pourra voir que les avantages de S3 Storage Lens limités à la portée de son compte, et non à celle de son organisation. Les tableaux de bord nécessitant un accès sécurisé ne seront plus mis à jour, mais ils pourront toujours interroger leurs données historiques, selon la période de disponibilité des données pour les requêtes.

Lorsqu'un compte en tant qu'administrateur délégué est supprimé, l'accès aux métriques du tableau de bord S3 Storage Lens par le propriétaire du compte est limité au niveau du compte uniquement. Les tableaux de bord organisationnels créés ne seront plus mis à jour, mais ils pourront toujours interroger leurs données historiques, selon la période de disponibilité des données pour les requêtes.

> **Note**
>
> - La désactivation de l'accès sécurisé désactive également automatiquement tous les tableaux de bord au niveau de l'organisation car S3 Storage Lens ne dispose plus d'un accès sécurisé aux comptes de l'organisation pour collecter et agréger des métriques de stockage.
> - Les comptes de gestion et d'administrateur délégué peuvent toujours voir les données historiques des tableaux de bord désactivés et ils peuvent interroger ces données tant qu'elles sont disponibles.

Pour désactiver l'accès sécurisé de S3 Storage Lens

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, puis Organization settings (Paramètres de l'organisation).
3. Dans Accès aux organisations, sélectionnez Modifier.

   La page Accès aux organisations s'ouvre. Ici, vous pouvez sélectionner Disable trusted access (Désactiver l'accès sécurisé) pour S3 Storage Lens.

Enregistrer des administrateurs délégués pour S3 Storage Lens

Une fois l'accès sécurisé activé, vous pouvez enregistrer un accès administrateur délégué aux comptes de votre organisation. Lorsqu'un compte est enregistré en tant qu'administrateur délégué, il reçoit l'autorisation d'accéder à toutes les opérations d' AWS Organizations API en lecture seule. Cela donne une visibilité aux membres et aux structures de votre organisation, qui peuvent créer des tableaux de bord S3 Storage Lens en votre nom.

Pour enregistrer des administrateurs délégués pour S3 Storage Lens

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, puis Organization settings (Paramètres de l'organisation).
3. Dans la section Delegated access (Accès délégué), pour Accounts (Comptes), sélectionnez Add account (Ajouter un compte).

   La page Delegated admin access (Accès administrateur délégué) s'ouvre. Ici, vous pouvez ajouter un ID de Compte AWS en tant qu'administrateur délégué pour créer des tableaux de bord au niveau de l'organisation pour tous les comptes et le stockage de votre organisation.

Annuler l'enregistrement d'administrateurs délégués pour S3 Storage Lens

Vous pouvez annuler l'enregistrement de l'accès d'un administrateur délégué aux comptes de votre organisation. Lorsqu'un compte est désenregistré en tant qu'administrateur délégué, il perd l'autorisation d'accéder à toutes les opérations d' AWS Organizations API en lecture seule qui fournissent de la visibilité aux membres et aux structures de votre organisation.

> **Note**
>
> - L'annulation de l'enregistrement d'un administrateur délégué désactive également automatiquement tous les tableaux de bord au niveau de l'organisation créés par l'administrateur délégué.
> - Les comptes d'administrateur délégué peuvent toujours voir les données historiques des tableaux de bord désactivés selon leurs périodes respectives de disponibilité pour les requêtes.

Pour annuler l'enregistrement des comptes pour l'accès d'un administrateur délégué

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Storage Lens, puis Organization settings (Paramètres de l'organisation).
3. Dans la section Accounts with delegated access (Comptes avec accès délégué), sélectionnez l'ID de compte dont vous souhaitez annuler l'enregistrement, puis Sélectionnez Remove (Supprimer).

Exemples Amazon S3 Storage Lens utilisant AWS CLI

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3. Pour en savoir plus, consultez la section Évaluation de l'activité et de l'utilisation du stockage avec Amazon S3 Storage Lens.

Les exemples suivants montrent comment utiliser S3 Storage Lens avec l'AWS Command Line Interface.

Rubriques

Fichiers d'aide pour l'utilisation d'Amazon S3 Storage Lens

Utilisez les fichiers JSON suivants et ses entrées de clé pour vos exemples.

Exemple de configuration S3 Storage Lens dans JSON

Example **`config.json`**

Le fichier `config.json` contient les détails d'une configuration de métriques et recommandations avancées S3 Storage Lens au niveau de l'organisation. Pour utiliser l'exemple suivant, remplacez *`user input placeholders`* par vos propres informations.

> **Note**
>
> Des frais supplémentaires s'appliquent pour les métriques et recommandations avancées. Pour plus d'informations, consultez Métriques et recommandations avancées.

```
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
```

```
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true,
          "SelectionCriteria":{
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
      "arn:aws:s3:::source_bucket1"
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::destination-bucket-name", // The destination bucket for your
metrics export must be in the same Region as your S3 Storage Lens configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
```

```
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true
    }
  }
}
```

Exemple de configuration S3 Storage Lens avec des groupes Storage Lens dans JSON

Example **`config.json`**

Le fichier `config.json` contient les détails que vous souhaitez appliquer à votre configuration Storage Lens lorsque vous utilisez des groupes Storage Lens. Pour utiliser l'exemple, remplacez *`user input placeholders`* par vos propres informations.

Pour attacher tous les groupes Storage Lens à votre tableau de bord, mettez à jour votre configuration Storage Lens avec la syntaxe suivante :

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
    "StorageLensGroupLevel": {},
  "IsEnabled": true
}
```

Pour inclure uniquement deux groupes Storage Lens dans la configuration de votre tableau de bord Storage Lens (*`slg-1`* et *`slg-2`*), utilisez la syntaxe suivante :

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Include": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
    },
  "IsEnabled": true
}
```

Pour exclure uniquement certains groupes Storage Lens de la configuration de votre tableau de bord, utilisez la syntaxe suivante :

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
```

```
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Exclude": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
    },
  "IsEnabled": true
}
```

Exemple de configuration de balises S3 Storage Lens dans JSON

Example **`tags.json`**

Le fichier `tags.json` contient les étiquettes que vous souhaitez appliquer à votre configuration S3 Storage Lens. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
[
    {
        "Key": "key1",
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
```

Exemple de configuration d'autorisations IAM pour S3 Storage Lens

Example **`permissions.json`** : nom de tableau de bord spécifique

Cet exemple de politique montre un fichier `permissions.json` IAM de S3 Storage Lens avec un nom de tableau de bord spécifique spécifié. Remplacez *`value1`*, *`us-east-1`*, *`your-dashboard-name`* et *`example-account-id`* par vos propres valeurs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/your-
dashboard-name"
        }
    ]
}
```

Example **`permissions.json`** : aucun nom de tableau de bord spécifique

Cet exemple de politique montre un fichier `permissions.json` IAM de S3 Storage Lens sans nom de tableau de bord spécifique spécifié. Remplacez *`value1`*, *`us-east-1`* et *`example-account-id`* par vos propres valeurs.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/*"
        }
    ]
}
```

Utilisation des configurations Amazon S3 Storage Lens avec la AWS CLI

Vous pouvez utiliser AWS CLI pour répertorier, créer, obtenir et mettre à jour vos configurations S3 Storage Lens. Les exemples suivants utilisent des fichiers d'aide JSON pour les entrées de clavier. Pour utiliser ces exemples, remplacez *`user input placeholders`* par vos propres informations.

Création d'une configuration S3 Storage Lens

Example Création d'une configuration S3 Storage Lens

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Création d'une configuration S3 Storage Lens sans étiquettes

Example Création d'une configuration S3 Storage Lens sans étiquettes

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json
```

Obtenir une configuration S3 Storage Lens

Example Obtenir une configuration S3 Storage Lens

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1
```

Répertoriage des configurations S3 Storage Lens sans jeton suivant

Example Répertoriage des configurations S3 Storage Lens sans jeton suivant

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1
```

Répertorie les configurations S3 Storage Lens

Example Répertorie les configurations S3 Storage Lens

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-
east-1 --next-token=abcdefghij1234
```

Supprimer une configuration S3 Storage Lens

Example Supprimer une configuration S3 Storage Lens

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Ajout d'étiquettes à une configuration S3 Storage Lens

Example Ajout d'étiquettes à une configuration S3 Storage Lens

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Obtenir des balises pour une configuration S3 Storage Lens

Example Obtenir des balises pour une configuration S3 Storage Lens

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Supprimer des balises pour une configuration S3 Storage Lens

Example Supprimer des balises pour une configuration S3 Storage Lens

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Utilisation d'Amazon S3 Storage Lens avec des exemples AWS Organizations utilisant AWS CLI

Utilisez Amazon S3 Storage Lens pour collecter des métriques de stockage et des données d'utilisation pour tous les comptes qui font partie de votre hiérarchie AWS Organizations. Pour de plus amples informations, veuillez consulter Utilisation d'Amazon S3 Storage Lens avec AWS Organizations.

Activer l'accès sécurisé de S3 Storage Lens à Organizations

Example Activer l'accès sécurisé de S3 Storage Lens à Organizations

```
aws organizations enable-aws-service-access --service-principal storage-lens.s3.amazonaws.com
```

Désactiver l'accès sécurisé de S3 Storage Lens à Organizations

Example Désactiver l'accès sécurisé de S3 Storage Lens à Organizations

```
aws organizations disable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Enregistrer des administrateurs délégués Organizations pour S3 Storage Lens

Example Enregistrer des administrateurs délégués Organizations pour S3 Storage Lens

Pour utiliser cet exemple, remplacez *111122223333* par l'ID de Compte AWS approprié.

```
aws organizations register-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Annuler l'enregistrement des administrateurs délégués Organizations pour S3 Storage Lens

Example Annuler l'enregistrement des administrateurs délégués Organizations pour S3 Storage Lens

Pour utiliser cet exemple, remplacez *111122223333* par l'ID de Compte AWS approprié.

```
aws organizations deregister-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Exemples d'utilisation d'Amazon S3 Storage Lens à l'aide du kit SDK pour Java

S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3. Pour en savoir plus, consultez la section Évaluation de l'activité et de l'utilisation du stockage avec Amazon S3 Storage Lens.

Les exemples suivants montrent comment utiliser S3 Storage Lens avec le kit AWS SDK for Java.

Rubriques

- Utilisation des configurations Amazon S3 Storage Lens à l'aide du kit SDK pour Java

Utilisation des configurations Amazon S3 Storage Lens à l'aide du kit SDK pour Java

Vous pouvez utiliser le kit SDK pour Java pour répertorier, créer, obtenir et mettre à jour vos configurations S3 Storage Lens. Les exemples suivants utilisent des fichiers d'aide JSON pour les entrées de clavier.

Rubriques

- Créer et mettre à jour une configuration S3 Storage Lens
- Supprimer une configuration S3 Storage Lens
- Obtenir une configuration S3 Storage Lens
- Répertorie les configurations S3 Storage Lens
- Ajout d'étiquettes à une configuration S3 Storage Lens
- Obtenir des balises pour une configuration S3 Storage Lens
- Supprimer des balises pour une configuration S3 Storage Lens
- Mise à jour de la configuration S3 Storage Lens par défaut avec les métriques et recommandations avancées
- Attachement d'un groupe Storage Lens à un tableau de bord S3 Storage Lens
- Exemples d'utilisation d'Amazon S3 Storage Lens avec AWS Organizations utilisant le kit SDK pour Java

Créer et mettre à jour une configuration S3 Storage Lens

Example Créer et mettre à jour une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
```

```
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
```

```
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            Include include = new Include()
                    .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
                    .withRegions(Arrays.asList("us-west-2"));

            StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                    .withSSES3(new SSES3());
            S3BucketDestination s3BucketDestination = new S3BucketDestination()
                    .withAccountId(exportAccountId)
                    .withArn(exportBucketArn)
                    .withEncryption(exportEncryption)
                    .withFormat(exportFormat)
                    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
                    .withPrefix("Prefix");
            CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                    .withIsEnabled(true);
            StorageLensDataExport dataExport = new StorageLensDataExport()
                    .withCloudWatchMetrics(cloudWatchMetrics)
                    .withS3BucketDestination(s3BucketDestination);

            StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                    .withArn(awsOrgARN);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
```

```
                    .withInclude(include)
                    .withDataExport(dataExport)
                    .withAwsOrg(awsOrg)
                    .withIsEnabled(true);

            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Supprimer une configuration S3 Storage Lens

Example Supprimer une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfiguration(new
 DeleteStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Obtenir une configuration S3 Storage Lens

Example Obtenir une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
```

```
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationResult;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final StorageLensConfiguration configuration =
                    s3ControlClient.getStorageLensConfiguration(new
 GetStorageLensConfigurationRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getStorageLensConfiguration();

            System.out.println(configuration.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Répertorie les configurations S3 Storage Lens

Example Répertorie les configurations S3 Storage Lens

```
package aws.example.s3control;
```

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationEntry;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationsRequest;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class ListDashboard {

    public static void main(String[] args) {
        String sourceAccountId = "Source Account ID";
        String nextToken = "nextToken";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final List<ListStorageLensConfigurationEntry> configurations =
                    s3ControlClient.listStorageLensConfigurations(new
 ListStorageLensConfigurationsRequest()
                            .withAccountId(sourceAccountId)
                            .withNextToken(nextToken)
                    ).getStorageLensConfigurationList();

            System.out.println(configurations.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Ajout d'étiquettes à une configuration S3 Storage Lens

Example Ajout d'étiquettes à une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
 PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags)
```

```
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Obtenir des balises pour une configuration S3 Storage Lens

Example Obtenir des balises pour une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;
import
 com.amazonaws.services.s3control.model.GetStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();
```

```
            final List<StorageLensTag> s3Tags = s3ControlClient
                    .getStorageLensConfigurationTagging(new
 GetStorageLensConfigurationTaggingRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getTags();

            System.out.println(s3Tags.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Supprimer des balises pour une configuration S3 Storage Lens

Example Supprimer des balises pour une configuration S3 Storage Lens

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationTaggingRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
```

```
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfigurationTagging(new
 DeleteStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mise à jour de la configuration S3 Storage Lens par défaut avec les métriques et recommandations avancées

Example Mise à jour de la configuration S3 Storage Lens par défaut avec les métriques et recommandations avancées

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
```

```
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
 cannot be modified.
        String sourceAccountId = "Source Account ID";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
 PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
```

```
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

> **Note**
>
> Des frais supplémentaires s'appliquent pour les métriques et recommandations avancées. Pour plus d'informations, consultez Métriques et recommandations avancées.

Attachement d'un groupe Storage Lens à un tableau de bord S3 Storage Lens

Example Attachement de tous les groupes Storage Lens à un tableau de bord

L'exemple de kit SDK for Java suivant attache tous les groupes Storage Lens du compte *111122223333* au tableau de bord *DashBoardConfigurationId* :

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
```

```
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWithStorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel();

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
```

```
            e.printStackTrace();
        }
    }
}
```

Example Attachement de deux groupes Storage Lens à un tableau de bord

L'exemple AWS SDK for Java suivant attache deux groupes Storage Lens (*StorageLensGroupName1* et *StorageLensGroupName2*) au tableau de bord *ExampleDashboardConfigurationId*.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
 StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName2);
```

```
            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Attachement de tous les groupes Storage Lens avec des exclusions

L'exemple de kit SDK for Java suivant attache tous les groupes Storage Lens au tableau de bord *ExampleDashboardConfigurationId*, à l'exception des deux groupes spécifiés *StorageLensGroupName1* et *StorageLensGroupName2*) :

```
package aws.example.s3control;
```

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroupsExcluded {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
 StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName2);

            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);
```

```
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Exemples d'utilisation d'Amazon S3 Storage Lens avec AWS Organizations utilisant le kit SDK pour Java

Utilisez Amazon S3 Storage Lens pour collecter des métriques de stockage et des données d'utilisation pour tous les comptes qui font partie de votre hiérarchie AWS Organizations. Pour de plus amples informations, veuillez consulter Utilisation d'Amazon S3 Storage Lens avec AWS Organizations.

Rubriques

- Activer l'accès sécurisé de S3 Storage Lens à Organizations
- Désactiver l'accès sécurisé de S3 Storage Lens à Organizations
- Enregistrer des administrateurs délégués Organizations pour S3 Storage Lens
- Annuler l'enregistrement des administrateurs délégués Organizations pour S3 Storage Lens

Activer l'accès sécurisé de S3 Storage Lens à Organizations

Example Activer l'accès sécurisé de S3 Storage Lens à Organizations

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.EnableAWSServiceAccessRequest;

public class EnableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.enableAWSServiceAccess(new
 EnableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Désactiver l'accès sécurisé de S3 Storage Lens à Organizations

Example Désactiver l'accès sécurisé de S3 Storage Lens à Organizations

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DisableAWSServiceAccessRequest;

public class DisableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            // Make sure to remove any existing delegated administrator for S3 Storage
 Lens
            // before disabling access; otherwise, the request will fail.
            organizationsClient.disableAWSServiceAccess(new
 DisableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Enregistrer des administrateurs délégués Organizations pour S3 Storage Lens

Example Enregistrer des administrateurs délégués Organizations pour S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.RegisterDelegatedAdministratorRequest;

public class RegisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
 RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Annuler l'enregistrement des administrateurs délégués Organizations pour S3 Storage Lens

Example Annuler l'enregistrement des administrateurs délégués Organizations pour S3 Storage Lens

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.DeregisterDelegatedAdministratorRequest;

public class DeregisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
 DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Utilisation des groupes S3 Storage Lens

Un groupe Amazon S3 Storage Lens regroupe les métriques à l'aide de filtres personnalisés basés sur les métadonnées des objets. Les groupes Storage Lens vous permettent d'analyser en détail les caractéristiques de vos données, telles que la répartition des objets par âge, les types de fichiers les plus courants, etc. Par exemple, vous pouvez filtrer les métriques par balise d'objet pour identifier les jeux de données dont la croissance est la plus rapide, ou visualiser votre stockage en fonction de la taille et de l'âge des objets pour définir votre stratégie d'archivage de stockage. Ainsi, les groupes Amazon S3 Storage Lens vous aident à mieux comprendre et à optimiser votre stockage S3.

Lorsque vous utilisez des groupes Storage Lens, vous pouvez analyser et filtrer les métriques S3 Storage Lens à l'aide de métadonnées d'objets, telles que les préfixes, les suffixes, les balises d'objet, la taille ou l'âge de l'objet. Vous pouvez également appliquer une combinaison de ces filtres. Après avoir attaché votre groupe Storage Lens à votre tableau de bord S3 Storage Lens, vous pouvez consulter les métriques S3 Storage Lens agrégées par les groupes Amazon S3 Storage Lens directement dans votre tableau de bord.

Par exemple, vous pouvez également filtrer vos métriques par taille d'objet ou par tranche d'âge afin de déterminer quelle partie de votre espace de stockage est constituée de petits objets. Vous pouvez ensuite utiliser ces informations avec S3 Intelligent-Tiering ou S3 Lifecycle pour transférer les petits objets vers différentes classes de stockage afin d'optimiser les coûts et le stockage.

Rubriques

Fonctionnement des groupes S3 Storage Lens

Vous pouvez utiliser les groupes Storage Lens pour regrouper les métriques à l'aide de filtres personnalisés basés sur les métadonnées des objets. Lorsque vous définissez un filtre personnalisé, vous pouvez utiliser des préfixes, des suffixes, des balises d'objet, des tailles d'objets, leur âge ou une combinaison de ces filtres personnalisés. Lors de la création d'un groupe Storage Lens, vous pouvez également inclure un filtre ou plusieurs conditions de filtre. Pour spécifier plusieurs conditions de filtre, utilisez les opérateurs logiques `And` ou `Or`.

Lorsque vous créez et configurez un groupe Storage Lens, le groupe Storage Lens lui-même agit comme un filtre personnalisé dans le tableau de bord auquel vous attachez le groupe. Dans votre

tableau de bord, vous pouvez ensuite utiliser le filtre de groupe Storage Lens pour obtenir des métriques de stockage en fonction du filtre personnalisé que vous avez défini dans le groupe.

Pour consulter les données de votre groupe Storage Lens dans votre tableau de bord S3 Storage Lens, vous devez attacher le groupe au tableau de bord après l'avoir créé. Une fois que votre groupe Storage Lens est attaché à votre tableau de bord Storage Lens, celui-ci collecte les statistiques d'utilisation du stockage pendant 48 heures. Vous pouvez ensuite visualiser ces données dans le tableau de bord Storage Lens ou les exporter via une exportation de métriques. Si vous oubliez d'attacher un groupe Storage Lens à un tableau de bord, les données de votre groupe Storage Lens ne seront pas capturées ou ne s'afficheront nulle part.

> **Note**
>
> - Lorsque vous créez un groupe S3 Storage Lens, vous créez une ressource AWS. Par conséquent, chaque groupe Storage Lens possède son propre Amazon Resource Name (ARN), que vous pouvez spécifier lorsque vous l'attachez ou que vous l'excluez d'un tableau de bord S3 Storage Lens.
> - Si votre groupe Storage Lens n'est pas attaché à un tableau de bord, vous n'aurez pas à payer de frais supplémentaires pour créer un groupe Storage Lens.
> - S3 Storage Lens regroupe les métriques d'utilisation d'un objet dans tous les groupes Storage Lens correspondants. Par conséquent, si un objet correspond aux conditions de filtre pour au moins deux groupes Storage Lens, vous constaterez des décomptes répétés pour le même objet dans l'ensemble de votre utilisation du stockage.

Vous pouvez créer un groupe Storage Lens au niveau du compte dans une région d'origine spécifiée (à partir de la liste des Régions AWS prises en charge). Vous pouvez ensuite attacher votre groupe Storage Lens à plusieurs tableaux de bord Storage Lens, à condition que les tableaux de bord se trouvent dans le même Compte AWS et la même région d'origine. Vous pouvez créer jusqu'à 50 groupes Storage Lens par région d'origine dans chaque Compte AWS.

Vous pouvez créer et gérer des groupes S3 Storage Lens à l'aide de la console Amazon S3, d'AWS Command Line Interface (AWS CLI), des kits SDK AWS ou de l'API REST Amazon S3.

Rubriques

- Configuration des groupes Storage Lens
- Balises de ressource AWS
- Exportation de métriques des groupes Storage Lens

Affichage des métriques agrégées d'un groupe Storage Lens

Vous pouvez consulter les métriques agrégées de vos groupes Storage Lens en attachant les groupes à un tableau de bord. Les groupes Storage Lens que vous souhaitez attacher doivent résider dans la région d'origine désignée dans le compte du tableau de bord.

Pour attacher un groupe Storage Lens à un tableau de bord, vous devez le spécifier dans la section Regroupement des groupes Storage Lens de la configuration de votre tableau de bord. Si vous avez plusieurs groupes Storage Lens, vous pouvez filtrer les résultats du Regroupement des groupes Storage Lens de sorte à inclure ou exclure les groupes de votre choix. Pour plus d'informations sur l'attachement de groupes à vos tableaux de bord, consultez the section called “Attachement ou retrait d'un groupe Storage Lens”.

Après avoir attaché vos groupes, vous verrez les données d'agrégation supplémentaires du groupe Storage Lens dans votre tableau de bord dans un délai de 48 heures.

> **Note**
>
> Pour consulter les métriques agrégées de votre groupe Storage Lens, vous devez attacher le groupe à un tableau de bord S3 Storage Lens.

Autorisations pour les groupes Storage Lens

Les groupes Storage Lens nécessitent certaines autorisations dans AWS Identity and Access Management (IAM) pour autoriser l'accès aux actions de groupe S3 Storage Lens. Pour accorder ces autorisations, vous pouvez utiliser une politique IAM basée sur l'identité. Vous pouvez attacher cette politique aux utilisateurs, groupes ou rôles IAM pour leur accorder des autorisations. Ces autorisations peuvent inclure la possibilité de créer ou de supprimer des groupes Storage Lens, d'afficher leurs configurations ou de gérer leurs balises.

L'utilisateur ou le rôle IAM auquel vous accordez des autorisations doit appartenir au compte qui a créé ou qui possède le groupe Storage Lens.

Pour utiliser les groupes Storage Lens et consulter les métriques de vos groupes Storage Lens, vous devez d'abord disposer des autorisations appropriées pour utiliser S3 Storage Lens. Pour plus d'informations, consultez the section called “Autorisations S3 Storage Lens”.

Pour créer et gérer des groupes S3 Storage Lens, vous devez disposer des autorisations IAM suivantes, en fonction des opérations que vous souhaitez effectuer :

Action	Autorisations IAM
Créer un nouveau groupe Storage Lens	`s3:CreateStorageLensGroup`
Créer un nouveau groupe Storage Lens avec des balises	`s3:CreateStorageLensGroup` , `s3:TagResource`
Mettre à jour un groupe Storage Lens existant	`s3:UpdateStorageLensGroup`
Renvoyer les détails de la configuration d'un groupe Storage Lens	`s3:GetStorageLensGroup`
Répertorier tous les groupes Storage Lens de votre région d'origine	`s3:ListStorageLensGroups`
Supprimer un groupe Storage Lens	`s3:DeleteStorageLensGroup`
Répertorier les balises qui ont été ajoutées à votre groupe Storage Lens	`s3:ListTagsForResource`
Ajouter ou mettre à jour une balise de groupe Storage Lens pour un groupe Storage Lens existant	`s3:TagResource`
Supprimer une balise d'un groupe Storage Lens	`s3:UntagResource`

Voici un exemple de configuration de votre politique IAM dans le compte qui crée le groupe Storage Lens. Pour utiliser cette politique, remplacez *`us-east-1`* par la région d'origine dans laquelle se trouve votre groupe Storage Lens. Remplacez *`111122223333`* par votre ID de Compte AWS, puis remplacez *`example-storage-lens-group`* par le nom de votre groupe Storage Lens. Pour appliquer ces autorisations à tous les groupes Storage Lens, remplacez *`example-storage-lens-group`* par *`*`*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EXAMPLE-Statement-ID",
            "Effect": "Allow",
            "Action": [
                "s3:CreateStorageLensGroup",
                "s3:UpdateStorageLensGroup",
                "s3:GetStorageLensGroup",
                "s3:ListStorageLensGroups",
                "s3:DeleteStorageLensGroup,
                "s3:TagResource",
                "s3:UntagResource",
                "s3:ListTagsForResource"
                ],
            "Resource": "arn:aws:s3:us-east-1:111122223333:storage-lens-group/example-storage-lens-group"
        }
    ]
}
```

Pour plus d'informations sur les autorisations S3 Storage Lens, consultez Autorisations Amazon S3 Storage Lens. Pour plus d'informations sur le langage des politiques IAM, consultez Politiques et autorisations dans Amazon S3.

Configuration des groupes Storage Lens

Nom de groupe S3 Storage Lens

Nous vous recommandons de donner à vos groupes Storage Lens des noms indiquant leur objectif afin que vous puissiez facilement déterminer les groupes que vous souhaitez attacher à vos tableaux de bord. Pour attacher un groupe Storage Lens à un tableau de bord, vous devez spécifier le groupe dans la section Regroupement des groupes Storage Lens de la configuration de votre tableau de bord.

Les noms de groupe Storage Lens doivent être uniques au sein du compte. Ils ne doivent pas dépasser 64 caractères et peuvent uniquement contenir des lettres (a à z, A à Z), des chiffres (0 à 9), des traits d'union (-) et des traits de soulignement (_).

Région d'accueil

La région d'origine est la Région AWS dans laquelle votre groupe Storage Lens est créé et géré. Votre groupe Storage Lens est créé dans la même région d'origine que votre tableau de bord Amazon S3 Storage Lens. La configuration et les métriques du groupe Storage Lens sont également stockées dans cette région. Vous pouvez créer jusqu'à 50 groupes Storage Lens dans une région d'origine.

Après avoir créé votre groupe Storage Lens, vous ne pouvez pas modifier la région d'origine.

Portée

Pour inclure des objets dans votre groupe Storage Lens, ils doivent être dans la portée de votre tableau de bord Amazon S3 Storage Lens. La portée de votre tableau de bord Storage Lens est déterminée par les compartiments que vous avez inclus dans la Portée du tableau de bord de la configuration de votre tableau de bord S3 Storage Lens.

Vous pouvez utiliser différents filtres pour vos objets afin de définir la portée de votre groupe Storage Lens. Pour afficher ces métriques de groupe Storage Lens dans votre tableau de bord S3 Storage Lens, les objets doivent correspondre aux filtres que vous incluez dans vos groupes Storage Lens. Supposons, par exemple, que votre groupe Storage Lens inclut des objets dotés du préfixe `marketing` et du suffixe `.png`, mais qu'aucun objet ne correspond à ces critères. Dans ce cas, les métriques de ce groupe Storage Lens ne seront pas générées lors de votre exportation de métriques quotidienne, et aucune métrique de ce groupe ne sera visible dans votre tableau de bord.

Filtres

Vous pouvez utiliser les filtres suivants dans un groupe S3 Storage Lens :

- Préfixes : indique le préfixe des objets inclus, qui est une chaîne de caractères au début du nom de la clé d'objet. Par exemple, la valeur `images` pour le filtre Préfixes inclut les objets dotés de l'un des préfixes suivants : `images/`, `images-marketing` et `images/production`. La longueur maximale d'un préfixe est de 1 024 octets.
- Suffixes : indique le suffixe des objets inclus (par exemple, `.png`, `.jpeg` ou `.csv`). La longueur maximale d'un suffixe est de 1 024 octets.
- Balises d'objet : indique la liste des balises d'objet sur lesquelles vous souhaitez appliquer le filtre. Une clé de balise ne peut pas dépasser 128 caractères Unicode et une valeur de balise ne peut pas dépasser 256 caractères Unicode. Notez que si le champ de valeur de balise d'objet est laissé vide, les groupes S3 Storage Lens ne font correspondre l'objet qu'aux autres objets dont les valeurs de balise sont également vides.

- Age : indique la tranche d'âge des objets inclus en jours. Seuls les entiers sont pris en charge.
- Taille : indique la plage de tailles des objets inclus en octets. Seuls les entiers sont pris en charge. La valeur maximale autorisée est de 5 To.

Balises d'objet d'un groupe Storage Lens

Vous pouvez créer un groupe Storage Lens comprenant jusqu'à 10 filtres de balises d'objet. L'exemple suivant inclut deux paires clé-valeur de balise d'objet en tant que filtres pour un groupe Storage Lens nommé *`Marketing-Department`*. Pour utiliser cet exemple, remplacez *`Marketing-Department`* par le nom de votre groupe, puis remplacez *`object-tag-key-1`*, *`object-tag-value-1`*, etc. par les paires clé-valeur de balise d'objet sur lesquelles vous souhaitez appliquer le filtre.

```
{
    "Name": "Marketing-Department",
    "Filter": {
     "MatchAnyTag":[
                {
                    "Key": "object-tag-key-1",
                    "Value": "object-tag-value-1"
                },
                {
                    "Key": "object-tag-key-2",
                    "Value": "object-tag-value-2"
                }
            ]
    }
}
```

Opérateurs logiques (**`And`** ou **`Or`**)

Pour inclure plusieurs conditions de filtre dans votre groupe Storage Lens, vous pouvez utiliser des opérateurs logiques (`And` ou `Or`). Dans l'exemple suivant, le groupe Storage Lens nommé *`Marketing-Department`* possède un opérateur `And` qui contient les filtres `Prefix`, `ObjectAge` et `ObjectSize`. Étant donné qu'un opérateur `And` est utilisé, seuls les objets répondant à toutes ces conditions de filtre seront inclus dans la portée du groupe Storage Lens.

Pour utiliser cet exemple, remplacez *`user input placeholders`* par les valeurs sur lesquelles vous souhaitez appliquer le filtre.

```
{
    "Name": "Marketing-Department",
    "Filter": {
        "And": {
            "MatchAnyPrefix": [
                "prefix-1",
                "prefix-2",
                "prefix-3/sub-prefix-1"
            ],
             "MatchObjectAge": {
                "DaysGreaterThan": 10,
                "DaysLessThan": 60
            },
            "MatchObjectSize": {
                "BytesGreaterThan": 10,
                "BytesLessThan": 60
            }
        }
    }
}
```

> **Note**
>
> Si vous souhaitez inclure des objets répondant à l'une des conditions des filtres, remplacez l'opérateur logique `And` par l'opérateur logique `Or` dans cet exemple.

Balises de ressource AWS

Chaque groupe S3 Storage Lens est considéré comme une ressource AWS ayant son propre Amazon Resource Name (ARN). Par conséquent, lorsque vous configurez votre groupe Storage Lens, vous pouvez éventuellement ajouter des balises de ressource AWS au groupe. Vous pouvez ajouter jusqu'à 50 balises par groupe Storage Lens. Pour créer un groupe Storage Lens avec des balises, vous devez disposer des autorisations `s3:CreateStorageLensGroup` et `s3:TagResource`.

Vous pouvez utiliser les balises de ressource AWS pour classer les ressources par département, secteur d'activité ou projet. Cela est utile lorsque vous avez de nombreuses ressources du même type. En appliquant des balises, vous pouvez rapidement identifier un groupe Storage Lens

spécifique en fonction des balises que vous lui avez affectées. Vous pouvez également utiliser des balises pour suivre et répartir les coûts.

En outre, lorsque vous ajoutez une balise de ressource AWS à votre groupe Storage Lens, vous activez le contrôle d'accès par attributs (ABAC). L'ABAC est une stratégie d'autorisation qui définit les autorisations en fonction des attributs, dans ce cas, des balises. Vous pouvez également utiliser des conditions qui spécifient les balises de ressource dans vos politiques IAM pour contrôler l'accès aux ressources AWS.

Vous pouvez modifier les clés et valeurs de balise, et vous pouvez retirer des balises d'une ressource à tout moment. En outre, tenez compte des limitations suivantes :

- Les clés de balise et valeurs de balise sont sensibles à la casse.
- Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur.
- Si vous supprimez une ressource, ses balises sont également supprimées.
- N'incluez pas de données privées ou sensibles dans vos balises de ressource AWS.
- Les balises système (balises dont les clés de balise commencent par `aws:`) ne sont pas prises en charge.
- La longueur de chaque clé de balise ne peut pas dépasser 128 caractères. La longueur de chaque valeur de balise ne peut pas dépasser 256 caractères.

Exportation de métriques des groupes Storage Lens

Les métriques de groupe S3 Storage Lens sont incluses dans l'exportation des métriques Amazon S3 Storage Lens pour le tableau de bord auquel le groupe Storage Lens est attaché. Pour obtenir des informations générales sur la fonctionnalité d'exportation des métriques Storage Lens, consultez Afficher les métriques Amazon S3 Storage Lens à l'aide d'une exportation de données.

Votre exportation de métriques pour les groupes Storage Lens incluent toute métrique S3 Storage Lens dans la portée du tableau de bord auquel le groupe Storage Lens est attaché. L'exportation inclut également des données de métriques supplémentaires pour les groupes Storage Lens.

Une fois que vous avez créé votre groupe Storage Lens, votre exportation de métriques est envoyée quotidiennement au compartiment que vous avez sélectionné lorsque vous avez configuré l'exportation de métriques pour le tableau de bord auquel votre groupe est attaché. La réception de la première exportation de métriques peut prendre jusqu'à 48 heures.

Pour générer des métriques lors de l'exportation quotidienne, les objets doivent correspondre aux filtres que vous incluez dans vos groupes Storage Lens. Si aucun objet ne correspond aux filtres que vous avez inclus dans votre groupe Storage Lens, aucune métrique ne sera générée. Toutefois, si un objet correspond à au moins deux groupes Storage Lens, l'objet est répertorié séparément pour chaque groupe lorsqu'il apparaît dans l'exportation de métriques.

Vous pouvez identifier les métriques pour les groupes Storage Lens en recherchant l'une des valeurs suivantes dans la colonne `record_type` de l'exportation de métriques pour votre tableau de bord :

- `STORAGE_LENS_GROUP_BUCKET`
- `STORAGE_LENS_GROUP_ACCOUNT`

La colonne `record_value` affiche l'ARN de ressource du groupe Storage Lens (par exemple, `arn:aws:s3:*us-east-1*:*111122223333*:storage-lens-group/*Marketing-Department*`).

Utilisation des groupes Storage Lens

Les groupes Amazon S3 Storage Lens regroupent les métriques à l'aide de filtres personnalisés basés sur les métadonnées des objets. Vous pouvez analyser et filtrer les métriques S3 Storage Lens à l'aide de préfixes, suffixes, balises d'objet, taille ou âge d'objet. Les groupes Amazon S3 Storage Lens vous permettent également de classer votre utilisation au sein des compartiments Amazon S3 et entre eux. Ainsi, vous serez en mesure de mieux comprendre et d'optimiser votre stockage S3.

Pour commencer à visualiser les données d'un groupe Storage Lens, vous devez d'abord attacher votre groupe Storage Lens à un tableau de bord S3 Storage Lens. Si vous devez gérer des groupes Storage Lens dans le tableau de bord, vous pouvez modifier la configuration du tableau de bord. Pour vérifier quels groupes Storage Lens sont associés à votre compte, vous pouvez les répertorier. Pour vérifier quels groupes Storage Lens sont attachés à votre tableau de bord, vous pouvez toujours consulter l'onglet Groupes Storage Lens du tableau de bord. Pour consulter ou mettre à jour la portée d'un groupe Storage Lens existant, vous pouvez consulter ses détails. Vous pouvez également supprimer définitivement un groupe Storage Lens.

Pour gérer les autorisations, vous pouvez créer et ajouter des balises de ressources AWS définies par l'utilisateur à vos groupes Storage Lens. Vous pouvez utiliser les balises de ressource AWS pour classer les ressources par département, secteur d'activité ou projet. Cela est utile lorsque vous avez de nombreuses ressources du même type. En appliquant des balises, vous pouvez rapidement identifier un groupe Storage Lens spécifique en fonction des balises que vous lui avez affectées.

En outre, lorsque vous ajoutez une balise de ressource AWS à votre groupe Storage Lens, vous activez le contrôle d'accès par attributs (ABAC). L'ABAC est une stratégie d'autorisation qui définit les autorisations en fonction des attributs, dans ce cas, des balises. Vous pouvez également utiliser des conditions qui spécifient les balises de ressource dans vos politiques IAM pour contrôler l'accès aux ressources AWS.

Rubriques

Création d'un groupe Storage Lens

Les exemples suivants montrent comment créer un groupe Amazon S3 Storage Lens à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour créer un groupe Storage Lens

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la barre de navigation en haut de la page, choisissez le nom de la AWS région actuellement affichée. Ensuite, choisissez la région vers laquelle vous souhaitez passer.
3. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
4. Choisissez Créer un groupe Storage Lens.
5. Sous Général, consultez votre région d'origine et entrez le nom de votre groupe Storage Lens.
6. Sous Portée, choisissez le filtre que vous souhaitez appliquer à votre groupe Storage Lens. Pour appliquer plusieurs filtres, choisissez vos filtres, puis l'opérateur logique AND ou OR.

- Pour le filtre Préfixes, choisissez Préfixes et entrez une chaîne de préfixe. Pour ajouter plusieurs préfixes, choisissez Ajouter un préfixe. Pour supprimer un préfixe, choisissez Supprimer en regard du préfixe que vous souhaitez supprimer.
- Pour le filtre Balises d'objet, choisissez Balises d'objet et entrez la paire clé-valeur de votre objet. Choisissez ensuite Ajouter une balise. Pour supprimer une balise, choisissez Supprimer en regard de la balise que vous souhaitez supprimer.
- Pour le filtre Suffixes, choisissez Suffixes et entrez une chaîne de suffixe. Pour ajouter plusieurs suffixes, choisissez Ajouter un suffixe. Pour supprimer un suffixe, choisissez Supprimer en regard du suffixe que vous souhaitez supprimer.
- Pour le filtre Age, spécifiez la tranche d'âge de l'objet en jours. Choisissez Spécifier l'âge minimum de l'objet, puis entrez l'âge minimal de l'objet. Choisissez ensuite Spécifier l'âge maximum de l'objet, puis entrez l'âge maximal de l'objet.
- Pour le filtre Taille, spécifiez la plage de tailles de l'objet et l'unité de mesure. Choisissez Spécifier la taille minimale d'objet, puis entrez la taille minimale de l'objet. Choisissez Spécifier la taille maximale d'objet, puis entrez la taille maximale de l'objet.

7. (Facultatif) Pour les balises de AWS ressources, ajoutez la paire clé-valeur, puis choisissez Ajouter une balise.
8. Choisissez Créer un groupe Storage Lens.

À l'aide du AWS CLI

L'exemple de AWS CLI commande suivant crée un groupe Storage Lens. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

L'exemple de AWS CLI commande suivant crée un groupe Storage Lens avec deux balises de AWS ressources. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json \
--tags Key=k1,Value=v1 Key=k2,Value=v2
```

Pour obtenir des exemples de configuration JSON, consultez Configuration des groupes Storage Lens.

Utilisation du AWS SDK pour Java

L' AWS SDK for Java exemple suivant crée un groupe Storage Lens. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

Example – Création d'un groupe Storage Lens avec un seul filtre

L'exemple suivant crée un groupe Storage Lens nommé *Marketing-Department*. Ce groupe possède un filtre d'âge d'objet qui spécifie la tranche d'âge sur *30* à *90* jours. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithObjectAge {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupFilter objectAgeFilter = StorageLensGroupFilter.builder()
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90)
                            .build())
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(objectAgeFilter)
                    .build();
```

```
            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example – Création d'un groupe Storage Lens avec un opérateur **AND** incluant plusieurs filtres

L'exemple suivant crée un groupe Storage Lens nommé *Marketing-Department*. Ce groupe utilise l'opérateur AND pour indiquer que les objets doivent correspondre à toutes les conditions de filtre. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.S3Tag;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupAndOperator;
```

```
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;


public class CreateStorageLensGroupWithAndFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create object tags.
            S3Tag tag1 = S3Tag.builder()
                    .key("object-tag-key-1")
                    .value("object-tag-value-1")
                    .build();
            S3Tag tag2 = S3Tag.builder()
                    .key("object-tag-key-2")
                    .value("object-tag-value-2")
                    .build();

            StorageLensGroupAndOperator andOperator =
 StorageLensGroupAndOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .matchAnyTag(tag1, tag2)
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90).build())
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L).build())
                    .build();

            StorageLensGroupFilter andFilter = StorageLensGroupFilter.builder()
                    .and(andOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(andFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
```

```
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example – Création d'un groupe Storage Lens avec un opérateur **OR** incluant plusieurs filtres

L'exemple suivant crée un groupe Storage Lens nommé *Marketing-Department*. Ce groupe utilise un opérateur OR pour appliquer un filtre de préfixe (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) ou un filtre de taille d'objet avec une plage de tailles comprise entre *1000* octets et *6000* octets. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupOrOperator;

public class CreateStorageLensGroupWithOrFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
```

```
        String accountId = "111122223333";

        try {
            StorageLensGroupOrOperator orOperator =
 StorageLensGroupOrOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L)
                            .build())
                    .build();

            StorageLensGroupFilter orFilter = StorageLensGroupFilter.builder()
                    .or(orOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(orFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Créez un groupe Storage Lens avec un seul filtre et deux balises de AWS ressources

L'exemple suivant crée un groupe Storage Lens nommé *Marketing-Department* qui possède un filtre de suffixe. Cet exemple ajoute également deux balises de AWS ressources au groupe Storage Lens. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.Tag;

public class CreateStorageLensGroupWithResourceTags {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create AWS resource tags.
            Tag resourceTag1 = Tag.builder()
                    .key("resource-tag-key-1")
                    .value("resource-tag-value-1")
                    .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-value-2")
                    .build();

            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();
```

```
            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Pour obtenir des exemples de configuration JSON, consultez Configuration des groupes Storage Lens.

Attachement ou retrait de groupes S3 Storage Lens à ou de votre tableau de bord

Une fois que vous êtes passé au niveau avancé dans Amazon S3 Storage Lens, vous pouvez attacher un groupe Storage Lens à votre tableau de bord. Si vous avez plusieurs groupes Storage Lens, vous pouvez inclure ou exclure les groupes de votre choix.

Vos groupes Storage Lens doivent résider dans la région d'origine désignée dans le compte du tableau de bord. Une fois que vous avez attaché un groupe Storage Lens à votre tableau de bord, vous recevrez les données d'agrégation supplémentaires du groupe Storage Lens dans votre exportation de métriques dans un délai de 48 heures.

> **Note**
>
> Si vous souhaitez consulter les métriques agrégées de votre groupe Storage Lens, vous devez les attacher à votre tableau de bord Storage Lens. Pour obtenir des exemples

de fichiers de configuration JSON pour le groupe Storage Lens, consultez Exemple de configuration S3 Storage Lens avec des groupes Storage Lens dans JSON.

Attachement d'un groupe Storage Lens à un tableau de bord S3 Storage Lens

Pour attacher un groupe Storage Lens à un tableau de bord Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, sous Storage Lens, choisissez Tableaux de bord.
3. Choisissez le bouton d'option du tableau de bord Storage Lens auquel vous souhaitez attacher un groupe Storage Lens.
4. Choisissez Edit (Modifier).
5. Sous Metrics selection (Sélection des métriques), choisissez Advanced metrics and recommendations (Métriques et recommandations avancées).
6. Sélectionnez Regroupement des groupes Storage Lens.

   > **Note**
   >
   > Par défaut, Métriques avancées est également sélectionné. Toutefois, vous pouvez également désélectionner ce paramètre, car il n'est pas obligatoire pour agréger les données des groupes Storage Lens.

7. Faites défiler la page jusqu'à Regroupement des groupes Storage Lens et spécifiez le ou les groupes Storage Lens que vous souhaitez inclure ou exclure dans l'agrégation de données. Vous pouvez utiliser les options de filtrage suivantes :

   - Si vous souhaitez inclure certains groupes Storage Lens, choisissez Inclure les groupes Storage Lens. Sous Groupes Storage Lens à inclure, sélectionnez vos groupes Storage Lens.
   - Si vous souhaitez inclure tous les groupes Storage Lens, sélectionnez Inclure tous les groupes Storage Lens de la région d'origine de ce compte.
   - Si vous souhaitez exclure certains groupes Storage Lens, choisissez Exclure les groupes Storage Lens. Sous Groupes Storage Lens à exclure, sélectionnez les groupes Storage Lens que vous souhaitez exclure.

8. Choisissez Enregistrer les modifications. Si vous avez configuré vos groupes Storage Lens correctement, vous verrez les données d'agrégation supplémentaires du groupe Storage Lens dans votre tableau de bord dans un délai de 48 heures.

Retrait d'un groupe Storage Lens d'un tableau de bord S3 Storage Lens

Pour retirer un groupe Storage Lens d'un tableau de bord S3 Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, sous Storage Lens, choisissez Tableaux de bord.
3. Choisissez le bouton d'option du tableau de bord Storage Lens duquel vous souhaitez retirer un groupe Storage Lens.
4. Choisissez Afficher la configuration du tableau de bord.
5. Choisissez Edit (Modifier).
6. Faites défiler la page jusqu'à la section Sélection des métriques.
7. Sous Regroupement des groupes Storage Lens, cliquez sur le X en regard du groupe Storage Lens que vous souhaitez retirer. Votre groupe Storage Lens est retiré.

   Si vous avez inclus tous vos groupes Storage Lens dans votre tableau de bord, décochez la case en regard de Inclure tous les groupes Storage Lens de la région d'origine de ce compte.
8. Choisissez Enregistrer les modifications.

> **Note**
>
> La répercussion des mises à jour de configuration dans votre tableau de bord prend jusqu'à 48 heures.

Visualisation des données de vos groupes Storage Lens

Vous pouvez visualiser les données de vos groupes Storage Lens en attachant le groupe à votre tableau de bord Amazon S3 Storage Lens. Une fois que vous avez inclus le groupe Storage Lens dans l'agrégation de groupes Storage Lens de la configuration de votre tableau de bord, l'affichage des données du groupe Storage Lens dans votre tableau de bord peut prendre jusqu'à 48 heures.

Une fois la configuration du tableau de bord mise à jour, tous les groupes Storage Lens récemment attachés apparaissent dans la liste des ressources disponibles sous l'onglet Groupes Storage Lens. Vous pouvez également analyser en détail l'utilisation du stockage dans votre onglet Aperçu en découpant les données selon une autre dimension. Par exemple, vous pouvez choisir l'un des éléments répertoriés dans les 3 principales catégories et choisir Analyser par pour découper les données selon une autre dimension. Vous ne pouvez pas appliquer la même dimension que le filtre lui-même.

> **Note**
>
> Vous ne pouvez pas appliquer un filtre de groupe Storage Lens en même temps qu'un filtre de préfixe, ou inversement. Vous ne pouvez pas non plus approfondir l'analyse d'un groupe Storage Lens à l'aide d'un filtre de préfixe.

Vous pouvez utiliser l'onglet Groupes Storage Lens du tableau de bord Amazon S3 Storage Lens pour personnaliser la visualisation des données pour les groupes Storage Lens attachés à votre tableau de bord. Vous pouvez visualiser les données de certains groupes Storage Lens attachés à votre tableau de bord ou de tous les groupes.

Lorsque vous visualisez les données d'un groupe Storage Lens dans votre tableau de bord S3 Storage Lens, tenez compte des points suivants :

- S3 Storage Lens regroupe les métriques d'utilisation d'un objet dans tous les groupes Storage Lens correspondants. Par conséquent, si un objet correspond aux conditions de filtre pour au moins deux groupes Storage Lens, vous constaterez des décomptes répétés pour le même objet dans l'ensemble de votre utilisation du stockage.
- Les objets doivent correspondre aux filtres que vous incluez dans vos groupes Storage Lens. Si aucun objet ne correspond aux filtres que vous incluez dans votre groupe Storage Lens, aucune métrique n'est générée. Pour déterminer s'il existe des objets non affectés, vérifiez le nombre total d'objets dans le tableau de bord au niveau du compte et au niveau du compartiment.

Mise à jour d'un groupe Storage Lens

Les exemples suivants montrent comment mettre à jour un groupe Amazon S3 Storage Lens. Vous pouvez mettre à jour un groupe Storage Lens à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour mettre à jour un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le groupe Storage Lens que vous souhaitez mettre à jour.
4. Sous Portée, choisissez Modifier.
5. Sur la page Portée, sélectionnez le filtre que vous souhaitez appliquer à votre groupe Storage Lens. Pour appliquer plusieurs filtres, sélectionnez vos filtres et choisissez l'opérateur logique AND ou OR.

   - Pour le filtre Préfixes, sélectionnez Préfixes et entrez une chaîne de préfixe. Pour ajouter plusieurs préfixes, choisissez Ajouter un préfixe. Pour supprimer un préfixe, choisissez Supprimer en regard du préfixe que vous souhaitez supprimer.
   - Pour le filtre Balises d'objet, entrez la paire clé-valeur de votre objet. Choisissez ensuite Ajouter une balise. Pour supprimer une balise, choisissez Supprimer en regard de la balise que vous souhaitez supprimer.
   - Pour le filtre Suffixes, sélectionnez Suffixes et entrez une chaîne de suffixe. Pour ajouter plusieurs suffixes, choisissez Ajouter un suffixe. Pour supprimer un suffixe, choisissez Supprimer en regard du suffixe que vous souhaitez supprimer.
   - Pour le filtre Age, spécifiez la tranche d'âge de l'objet en jours. Choisissez Spécifier l'âge minimum de l'objet, puis entrez l'âge minimal de l'objet. Pour Spécifier l'âge maximum de l'objet, entrez l'âge maximal de l'objet.
   - Pour le filtre Taille, spécifiez la plage de tailles de l'objet et l'unité de mesure. Choisissez Spécifier la taille minimale d'objet, puis entrez la taille minimale de l'objet. Pour Spécifier la taille maximale d'objet, entrez la taille maximale de l'objet.
6. Choisissez Enregistrer les modifications. La page de détails du groupe Storage Lens s'affiche.
7. (Facultatif) Si vous souhaitez ajouter une nouvelle balise de ressource AWS, faites défiler la page jusqu'à la section Balises de ressources AWS, puis choisissez Ajouter des balises. La page Ajouter des balises s'affiche.

   Ajoutez la nouvelle paire clé-valeur, puis choisissez Enregistrer les modifications. La page de détails du groupe Storage Lens s'affiche.

8. (Facultatif) Si vous souhaitez retirer une balise de ressource AWS existante, faites défiler la page jusqu'à la section Balises de ressources AWS, puis sélectionnez la balise de ressource. Ensuite, choisissez Supprimer. La boîte de dialogue Supprimer les balises AWS s'affiche.

   Choisissez à nouveau Supprimer pour supprimer définitivement la balise de ressource AWS.

   > **Note**
   >
   > Une fois que vous avez définitivement supprimé une balise de ressource AWS, elle ne peut pas être restaurée.

Utilisation de AWS CLI

L'exemple de commande AWS CLI suivant renvoie les détails de configuration d'un groupe Storage Lens nommé *marketing-department*. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

L'exemple AWS CLI suivant met à jour un groupe Storage Lens. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control update-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Pour obtenir des exemples de configuration JSON, consultez Configuration des groupes Storage Lens.

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant renvoie les détails de configuration du groupe Storage Lens *Marketing-Department* dans le compte *111122223333*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

L'exemple suivant met à jour le groupe Storage Lens nommé *Marketing-Department* dans le compte *111122223333*. Cet exemple met à jour la portée du tableau de bord de sorte à inclure les objets correspondant à l'un des suffixes suivants : *.png*, *.gif*, *.jpg* ou *.jpeg*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.UpdateStorageLensGroupRequest;

public class UpdateStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create updated filter.
            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg", ".jpeg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();

            UpdateStorageLensGroupRequest updateStorageLensGroupRequest =
 UpdateStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId)
                    .build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.updateStorageLensGroup(updateStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
```

```
        }
    }
}
```

Pour obtenir des exemples de configuration JSON, consultez Configuration des groupes Storage Lens.

Gestion des balises de ressource AWS avec les groupes Storage Lens

Chaque groupe Amazon S3 Storage Lens est considéré comme une ressource AWS ayant son propre Amazon Resource Name (ARN). Par conséquent, lorsque vous configurez votre groupe Storage Lens, vous pouvez éventuellement ajouter des balises de ressource AWS au groupe. Vous pouvez ajouter jusqu'à 50 balises par groupe Storage Lens. Pour créer un groupe Storage Lens avec des balises, vous devez disposer des autorisations `s3:CreateStorageLensGroup` et `s3:TagResource`.

Vous pouvez utiliser les balises de ressource AWS pour classer les ressources par département, secteur d'activité ou projet. Cela est utile lorsque vous avez de nombreuses ressources du même type. En appliquant des balises, vous pouvez rapidement identifier un groupe Storage Lens spécifique en fonction des balises que vous lui avez affectées. Vous pouvez également utiliser des balises pour suivre et répartir les coûts.

En outre, lorsque vous ajoutez une balise de ressource AWS à votre groupe Storage Lens, vous activez le contrôle d'accès par attributs (ABAC). L'ABAC est une stratégie d'autorisation qui définit les autorisations en fonction des attributs, dans ce cas, des balises. Vous pouvez également utiliser des conditions qui spécifient les balises de ressource dans vos politiques IAM pour contrôler l'accès aux ressources AWS.

Vous pouvez modifier les clés et valeurs d'identification, et vous pouvez retirer des identifications d'une ressource à tout moment. En outre, tenez compte des limitations suivantes :

- Les clés de balise et valeurs de balise sont sensibles à la casse.
- Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur.
- Si vous supprimez une ressource, les balises associées à celle-ci seront également supprimées.
- N'incluez pas de données privées ou sensibles dans vos balises de ressource AWS.
- Les balises système (dont les clés de balise commencent par `aws:`) ne sont pas prises en charge.
- La longueur de chaque clé de balise ne peut pas dépasser 128 caractères. La longueur de chaque valeur de balise ne peut pas dépasser 256 caractères.

Les exemples suivants montrent comment utiliser les balises de ressource AWS avec des groupes Storage Lens.

Rubriques

Ajout d'une balise de ressource AWS à un groupe Storage Lens

Les exemples suivants montrent comment utiliser des balises de ressource AWS à un groupe Amazon S3 Storage Lens. Vous pouvez ajouter des balises de ressource à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour ajouter une balise de ressource AWS à un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le groupe Storage Lens que vous souhaitez mettre à jour.
4. Sous Balises de ressources AWS, choisissez Ajouter des balises.
5. Sur la page Ajouter des balises, ajoutez la nouvelle paire clé-valeur.

   > **Note**
   >
   > L'ajout d'une nouvelle balise ayant la même clé qu'une balise existante remplace la valeur de balise précédente.

6. (Facultatif) Pour ajouter plusieurs nouvelles balises, choisissez à nouveau Ajouter une balise pour continuer à ajouter de nouvelles entrées. Vous pouvez ajouter jusqu'à 50 balises de ressources AWS à votre groupe Storage Lens.
7. (Facultatif) Si vous souhaitez retirer une entrée récemment ajoutée, choisissez Retirer en regard de la balise que vous souhaitez retirer.

8. Choisissez Enregistrer les modifications.

Utilisation de AWS CLI

L'exemple de commande AWS CLI suivant ajoute deux balises de ressources à un groupe Storage Lens existant nommé *marketing-department*. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v1 Key=k2,Value=v2
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant ajoute deux balises de ressources AWS à un groupe Storage Lens existant. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class TagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
```

```
                    .value("resource-tag-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mise à jour des valeurs de balise d'un groupe Storage Lens

Les exemples suivants montrent comment mettre à jour les valeurs de balise d'un groupe Storage Lens à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour mettre à jour une balise de ressource AWS pour un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le groupe Storage Lens que vous souhaitez mettre à jour.
4. Sous Balises de ressources AWS, sélectionnez la balise que vous souhaitez mettre à jour.

5. Ajoutez la nouvelle valeur de balise en utilisant la même clé que la paire clé-valeur que vous souhaitez mettre à jour. Cliquez sur l'icône représentant une coche pour mettre à jour la valeur de balise.

   > **Note**
   >
   > L'ajout d'une nouvelle balise ayant la même clé qu'une balise existante remplace la valeur de balise précédente.

6. (Facultatif) Si vous souhaitez ajouter de nouvelles balises, choisissez Ajouter une balise pour ajouter de nouvelles entrées. La page Ajouter des balises s'affiche.

   Vous pouvez ajouter jusqu'à 50 balises de ressources AWS pour votre groupe Storage Lens. Lorsque vous avez terminé d'ajouter des nouvelles balises, choisissez Enregistrer les modifications.

7. (Facultatif) Si vous souhaitez retirer une entrée récemment ajoutée, choisissez Retirer en regard de la balise que vous souhaitez retirer. Lorsque vous avez terminé de retirer des balises, choisissez Enregistrer les modifications.

Utilisation de AWS CLI

L'exemple de commande AWS CLI suivant met à jour deux valeurs de balise pour le groupe Storage Lens nommé *`marketing-department`*. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v3 Key=k2,Value=v4
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant met à jour deux valeurs de balise pour le groupe Storage Lens. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class UpdateTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag updatedResourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-updated-value-1")
                .build();
            Tag updatedResourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-updated-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(updatedResourceTag1, updatedResourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Suppression d'une balise de ressource AWS d'un groupe Storage Lens

Les exemples suivants montrent comment supprimer une balise de ressource AWS d'un groupe Storage Lens. Vous pouvez supprimer des balises à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour supprimer une balise de ressource AWS d'un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le groupe Storage Lens que vous souhaitez mettre à jour.
4. Sous Balises de ressources AWS, sélectionnez la paire clé-valeur que vous souhaitez supprimer.
5. Choisissez Supprimer. La boîte de dialogue Supprimer les balises de ressources AWS s'affiche.

> **Note**
>
> Si des balises sont utilisées pour contrôler l'accès, cette action peut affecter les ressources associées. Une fois que vous avez définitivement supprimé une balise, elle ne peut pas être restaurée.

6. Choisissez Supprimer pour supprimer définitivement la paire clé-valeur.

Utilisation de AWS CLI

La commande AWS CLI suivante supprime deux balises de ressources AWS d'un groupe Storage Lens existant. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control untag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-
Department \
--region us-east-1 --tag-keys k1 k2
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant supprime deux balises de ressources AWS de l'Amazon Resource Name (ARN) du groupe Storage Lens que vous spécifiez dans le compte *111122223333*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;

public class UntagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            String tagKey1 = "resource-tag-key-1";
            String tagKey2 = "resource-tag-key-2";
            UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tagKeys(tagKey1, tagKey2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.untagResource(untagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Liste des balises du groupe Storage Lens

Les exemples suivants montrent comment répertorier les balises de ressources AWS associées à un groupe Storage Lens. Vous pouvez répertorier les balises à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour consulter la liste des balises et des valeurs des balises d'un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le groupe Storage Lens qui vous intéresse.
4. Faites défiler la page jusqu'à la section Balises de ressources AWS. Toutes les balises de ressources AWS définies par l'utilisateur qui sont ajoutées à votre groupe Storage Lens sont répertoriées avec leurs valeurs de balise.

Utilisation de AWS CLI

L'exemple de commande AWS CLI suivant répertorie toutes les valeurs de balise du groupe Storage Lens nommé *marketing-department*. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control list-tags-for-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant répertorie les valeurs de balise du groupe Storage Lens pour l'Amazon Resource Name (ARN) du groupe Storage Lens que vous spécifiez. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;

public class ListTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            ListTagsForResourceRequest listTagsForResourceRequest =
 ListTagsForResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListTagsForResourceResponse response =
 s3ControlClient.listTagsForResource(listTagsForResourceRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Liste de tous les groupes Storage Lens

Les exemples suivants montrent comment répertorier tous les groupes Amazon S3 Storage Lens d'un Compte AWS et d'une région d'origine. Ces exemples montrent comment répertorier tous les groupes Storage Lens à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour répertorier tous les groupes Storage Lens d'un compte et d'une région d'origine

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, la liste des groupes Storage Lens de votre compte s'affiche.

Utilisation de AWS CLI

L'exemple AWS CLI suivant répertorie tous les groupes Storage Lens de votre compte. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \
--region us-east-1
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant répertorie les groupes Storage Lens du compte *111122223333*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsRequest;
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsResponse;

public class ListStorageLensGroups {
    public static void main(String[] args) {
        String accountId = "111122223333";

        try {
            ListStorageLensGroupsRequest listStorageLensGroupsRequest =
 ListStorageLensGroupsRequest.builder()
```

```
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListStorageLensGroupsResponse response =
 s3ControlClient.listStorageLensGroups(listStorageLensGroupsRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Affichage des détails d'un groupe Storage Lens

Les exemples suivants montrent comment afficher les détails de configuration d'un groupe Amazon S3 Storage Lens. Vous pouvez afficher ces détails à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour consulter les détails de configuration d'un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, choisissez le bouton d'option en regard du groupe Storage Lens qui vous intéresse.
4. Sélectionnez Afficher les détails. Vous pouvez désormais passer en revue les détails de votre groupe Storage Lens.

Utilisation de AWS CLI

L'exemple AWS CLI suivant renvoie les détails de configuration d'un groupe Storage Lens. Pour utiliser cet exemple de commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant renvoie les détails de configuration du groupe Storage Lens nommé *Marketing-Department* dans le compte *111122223333*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Suppression d'un groupe Storage Lens

Les exemples suivants montrent comment supprimer un groupe Amazon S3 Storage Lens à l'aide de la console Amazon S3, AWS Command Line Interface (AWS CLI) et AWS SDK for Java.

Utilisation de la console S3

Pour supprimer un groupe Storage Lens

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Groupes Storage Lens.
3. Sous Groupes Storage Lens, cliquez sur le bouton d'option en regard du groupe Storage Lens que vous souhaitez supprimer.
4. Choisissez Supprimer. La boîte de dialogue Supprimer le groupe Storage Lens s'affiche.
5. Choisissez à nouveau Supprimer pour supprimer définitivement votre groupe Storage Lens.

> **Note**
>
> Une fois que vous avez supprimé un groupe Storage Lens, il ne peut pas être restauré.

Utilisation de AWS CLI

L'exemple AWS CLI suivant supprime le groupe Storage Lens nommé *`marketing-department`*. Pour utiliser cet exemple de commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Utilisation du kit AWS SDK pour Java

L'exemple AWS SDK for Java suivant supprime le groupe Storage Lens nommé *Marketing-Department* dans le compte *111122223333*. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.DeleteStorageLensGroupRequest;

public class DeleteStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            DeleteStorageLensGroupRequest deleteStorageLensGroupRequest =
 DeleteStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.deleteStorageLensGroup(deleteStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
```

```
}
```

Suivi des demandes Amazon S3 à l'aide d' AWS X-Ray

AWS X-Ray recueille des données sur les demandes que sert votre application. Vous pouvez ensuite afficher et filtrer les données pour identifier et résoudre les problèmes de performances et les erreurs dans vos applications distribuées et votre architecture de micro-services. Pour toutes les demandes suivies transmises à votre application, X-Ray vous affiche les informations détaillées sur la demande et la réponse, mais également sur les appels que votre application effectue vers des bases de données, micro-services, ressources AWS en aval et API Web HTTP.

Pour plus d'informations, veuillez consulter Présentation de AWS X-Ray dans le Manuel du développeur AWS X-Ray.

Rubriques

- Fonctionnement de X-Ray avec Amazon S3
- Régions disponibles

Fonctionnement de X-Ray avec Amazon S3

AWS X-Ray prend en charge la propagation du contexte de suivi pour Amazon S3, de sorte que vous pouvez visualiser les demandes de bout en bout au fur et à mesure qu'elles transitent dans l'ensemble de votre application. X-Ray regroupe les données générées par les différents services tels qu'Amazon S3, AWS Lambda et Amazon EC2, ainsi que les nombreuses ressources qui composent votre application. Il vous fournit une vue d'ensemble des performances de votre application.

Amazon S3 s'intègre à X-Ray pour propager le contexte de suivi et vous donner une chaîne de demandes avec des nœuds en amont et en aval. Si un service en amont inclut un en-tête de suivi au format valide avec sa demande S3, Amazon S3 transmet l'en-tête de suivi lors de la remise des notifications d'événements à des services en aval tels que Lambda, Amazon SQS et Amazon SNS. Si tous ces services sont activement intégrés à X-Ray, ils sont liés en une seule chaîne de demandes pour vous fournir tous les détails de vos demandes Amazon S3.

Pour envoyer des en-têtes de suivi X-Ray via Amazon S3, vous devez inclure un X-Amzn-Trace-ID formaté dans vos demandes. Vous pouvez également instrumenter le client Amazon S3 à l'aide des kits SDK AWS X-Ray. Pour obtenir la liste des kits SDK pris en charge, consultez la documentation AWS X-Ray.

Cartes de service

Les cartes de service X-Ray vous montrent les relations entre Amazon S3 et les autres services et ressources AWS de votre application, le tout presque en temps réel. Pour afficher les demandes de bout en bout à l'aide des cartes de service X-Ray, vous pouvez utiliser la console X-Ray pour afficher une carte des connexions entre Amazon S3 et d'autres services que votre application utilise. Vous pouvez facilement détecter le point d'origine d'une latence élevée, visualiser la répartition des nœuds de ces services, puis explorer plus en profondeur les services et les chemins spécifiques affectant les performances de l'application.

X-Ray Analytics

Vous pouvez également utiliser la console X-Ray Analytics pour analyser les traces, afficher les métriques telles que la latence et les taux d'échec, et générer des informations pour vous aider à identifier et à résoudre les problèmes. Cette console affiche également des métriques telles que la latence moyenne et les taux d'échec. Pour plus d'informations, consultez Console AWS X-Ray dans le Guide du développeur AWS X-Ray.

Régions disponibles

La prise en charge d'AWS X-Ray pour Amazon S3 est disponible dans toutes les régions AWS X-Ray. Pour plus d'informations, consultez Amazon S3 et AWS X-Ray dans le Guide du développeur AWS X-Ray.

Hébergement d'un site Web statique à l'aide d'Amazon S3

Vous pouvez utiliser Amazon S3 pour héberger un site web statique. Sur un site web statique, les pages web individuelles contiennent du contenu statique. Elles peuvent également contenir des scénarios côté client.

Par contre, un site web dynamique repose sur un traitement côté serveur, comprenant des scripts côté serveur tels que PHP, JSP ou ASP.NET. Amazon S3 ne prend pas en charge les scripts côté serveur, mais AWS dispose d'autres ressources pour héberger des sites Web dynamiques. Pour en savoir plus sur l'hébergement de sites Web sur AWS, consultez la section Hébergement Web.

> **Note**
>
> Vous pouvez utiliser la AWS Amplify console pour héberger une application Web d'une seule page. La console AWS Amplify prend en charge les applications d'une seule page créées avec des frameworks d'applications d'une seule page (par exemple, React JS, Vue JS, Angular JS et Nuxt) et des générateurs de site statique (par exemple, Gatsby JS, React-static, Jekyll et Hugo). Pour plus d'informations, consultez Prise en main dans le Guide de l'utilisateur de la console AWS Amplify .
> Les points de terminaison de site web Amazon S3 ne prennent pas en charge HTTPS. Si vous souhaitez utiliser le protocole HTTPS, vous pouvez utiliser Amazon CloudFront pour diffuser un site Web statique hébergé sur Amazon S3. Pour plus d'informations, consultez Comment utiliser CloudFront pour répondre aux requêtes HTTPS pour mon compartiment Amazon S3 ? Pour utiliser le protocole HTTPS avec un domaine personnalisé, veuillez consulter Configuration d'un site web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

Pour plus d'informations sur l'hébergement d'un site Web statique sur Amazon S3, y compris les instructions et les step-by-step procédures pas à pas, consultez les rubriques suivantes.

Rubriques

- Définition des autorisations pour l'accès au site web
- (Facultatif) Journalisation du trafic web
- (Facultatif) Configuration de la redirection de pages web

Points de terminaison de sites web

Lorsque vous configurez votre compartiment en tant que site web statique, il est disponible au point de terminaison du site web spécifique de la Région AWS du compartiment. Les points de terminaison de sites Web sont différents de ceux auxquels vous adressez des demandes REST API. Pour plus d'informations sur les différences entre les points de terminaison, consultez Différences clés entre un point de terminaison de site web et un point de terminaison de l'API REST.

En fonction de votre Région, le point de terminaison de votre site web Amazon S3 respecte l'un des deux formats suivants.

- s3-siteweb tiret (-) Région - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-siteweb point (.) Région - `http://bucket-name.s3-website.Region.amazonaws.com`

Ces URL renvoient le document d'index par défaut que vous avez configuré pour le site web : Pour obtenir la liste complète des points de terminaison de sites web Amazon S3, veuillez consulter Points de terminaison de site web Amazon S3.

> **Note**
>
> Pour renforcer la sécurité de vos sites Web statiques Amazon S3, les domaines de point de terminaison des sites Web Amazon S3 (par exemple, s3- website-us-east -1.amazonaws.com ou s3-website.ap-south-1.amazonaws.com) sont enregistrés dans la liste des suffixes publics (PSL). Pour plus de sécurité, nous vous recommandons d'utiliser des cookies avec un préfixe `__Host-` si vous devez définir des cookies sensibles dans le nom de domaine de vos sites web statiques Amazon S3. Cette pratique vous aidera à protéger votre domaine contre les tentatives de falsification de requêtes intersites (CSRF). Pour plus d'informations, consultez la page Set-Cookie du Mozilla Developer Network.

Si vous souhaitez que votre site web soit public, vous devez rendre l'ensemble de votre contenu accessible publiquement en lecture pour que vos clients puissent accéder au point de terminaison du site web. Pour plus d'informations, consultez Définition des autorisations pour l'accès au site web.

> **Important**
>
> Les points de terminaison de site web Amazon S3 ne prennent pas en charge le protocole HTTPS ou les points d'accès. Si vous souhaitez utiliser le protocole HTTPS, vous pouvez utiliser Amazon CloudFront pour diffuser un site Web statique hébergé sur Amazon S3. Pour plus d'informations, consultez Comment utiliser CloudFront pour répondre aux requêtes HTTPS pour mon compartiment Amazon S3 ? Pour utiliser le protocole HTTPS avec un domaine personnalisé, veuillez consulter Configuration d'un site web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.
>
> Les compartiments de type Paiement par le demandeur ne permettent pas l'accès via un point de terminaison de site web. Toute demande à un compartiment de type reçoit une réponse 403 Accès refusé. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.

Rubriques

Exemples de point de terminaison de site web

Les exemples suivants montrent comment accéder à un compartiment Amazon S3 configuré en tant que site web statique.

Example - Demande d'un objet au niveau racine

Pour demander un objet spécifique stocké au niveau racine dans le compartiment, utilisez la structure d'URL suivante.

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Par exemple, l'URL suivante demande l'objet `photo.jpg` stocké au niveau racine dans le compartiment.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example - Demande d'un objet dans un préfixe

Pour demander un objet stocké dans un dossier de votre compartiment, utilisez la structure d'URL suivante.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

L'URL suivante demande l'objet `docs/doc1.html` dans votre compartiment.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Ajout d'un DNS CNAME

Si vous avez un domaine enregistré, vous pouvez ajouter une entrée DNS CNAME pour diriger vers le point de terminaison du site Web Amazon S3. A titre d'exemple, si vous avez enregistré le domaine `www.example-bucket.com`, vous pouvez créer un compartiment `www.example-bucket.com` et ajouter un enregistrement DNS CNAME qui pointe vers `www.example-bucket.com.s3-website.Region.amazonaws.com`. Toutes les demandes adressées à `http://www.example-bucket.com` sont réacheminées vers `www.example-bucket.com.s3-website.Region.amazonaws.com`.

Pour plus d'informations, consultez Personnalisation des URL Amazon S3 avec des enregistrements CNAME.

Utilisation d'un domaine personnalisé avec Route 53

Au lieu d'accéder au site web à l'aide d'un point de terminaison de site web Amazon S3, vous pouvez utiliser votre propre domaine enregistré auprès d'Amazon Route 53 pour diffuser votre contenu (par exemple,, `example.com`. Vous pouvez utiliser Amazon S3 avec Route 53 pour héberger un site web sur le domaine racine. À titre d'exemple, si votre domaine racine est `example.com` et que votre site web est hébergé sur Amazon S3, les visiteurs peuvent accéder à votre site web à partir de leur navigateur en entrant `http://www.example.com` ou `http://example.com`.

Pour afficher un exemple de procédure, veuillez consulter Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

Différences clés entre un point de terminaison de site web et un point de terminaison de l'API REST

Un point de terminaison de site web Amazon S3 est optimisé pour un accès depuis un navigateur web. Le tableau suivant résume les principales différences entre un point de terminaison de l'API REST et un point de terminaison de site web.

Principales différences	point de terminaison de l'API REST	point de terminaison des sites Web
Contrôle d'accès	Prend en charge les contenus public et privé	Prend en charge uniquement le contenu public
Gestion des messages d'erreur	Renvoie une réponse au format XML	Renvoie un document HTML
Prise en charge de redirection	Ne s'applique pas	Prend en charge les redirections au niveau de l'objet et du compartiment
Demandes prises en charge	Prend en charge les opérations relatives aux compartiments et aux objets.	Prend en charge uniquement les demandes GET et HEAD sur les objets
Réponses aux demandes GET et HEAD à la racine du compartiment	Renvoie une liste de clés d'objet dans le compartiment	Renvoie le document d'index qui est spécifié dans la configuration du site Web
Prise en charge du protocole SSL (Secure Sockets Layer)	Prend en charge toutes les connexions SSL	Ne prend pas en charge les connexions SSL

Pour obtenir la liste complète des points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans la Références générales AWS.

Activation de l'hébergement de sites web

Lorsque vous configurez un compartiment en tant que site Web statique, vous devez activer l'hébergement de sites Web statiques, configurer un document d'index et définir des autorisations.

Vous pouvez activer l'hébergement statique de sites Web à l'aide de la console Amazon S3, de l'API REST, AWS des SDK, du AWS CLI, ou AWS CloudFormation.

Pour configurer votre site Web avec un domaine personnalisé, veuillez consulter Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

Utiliser la console S3.

Pour activer l'hébergement de site Web statique

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment pour lequel vous souhaitez activer l'hébergement de sites web statiques.
3. Choisissez Propriétés.
4. Sous Static website hosting (Hébergement de site Web statique), choisissez Edit (Modifier).
5. Choisissez Utiliser ce compartiment pour héberger un site Web.
6. Sous Static website hosting (Hébergement de site web statique), choisissez Enable (Activer).
7. Dans Index document (Document d'index), entrez le nom du document d'index, généralement `index.html`.

   Le nom du document d'index est sensible à la casse et doit correspondre exactement au nom de fichier du document d'index HTML que vous prévoyez de charger dans votre compartiment S3. Lorsque vous configurez un compartiment pour l'hébergement d'un site web, vous devez indiquer un document d'index. Amazon S3 renvoie ce document d'index lorsque des demandes sont faites dans le domaine racine ou dans n'importe quel sous-dossier. Pour plus d'informations, consultez Configuration d'un document d'index.

8. Pour fournir votre propre document d'erreur personnalisé pour les erreurs de classe 4XX, entrez le nom du fichier du document d'erreur personnalisé dans Document d'erreur.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom de fichier du document d'erreur HTML que vous prévoyez de charger dans votre compartiment S3. Si vous ne spécifiez pas de document d'erreur personnalisé et qu'une erreur se produit, Amazon S3 renvoie un document d'erreur HTML par défaut. Pour plus d'informations, consultez Configuration d'un document d'erreur personnalisé.

9. (Facultatif) Si vous souhaitez spécifier des règles de redirection avancées, décrivez les règles à l'aide du langage JSON dans Redirection rules (Règles de redirection).

   Par exemple, vous pouvez acheminer les demandes de façon conditionnelle en fonction des noms ou préfixes de clés d'objets dans la demande. Pour plus d'informations, consultez Configurer des règles de redirection pour utiliser des redirections conditionnelles avancées.

10. Choisissez Enregistrer les modifications.

    Amazon S3 permet l'hébergement de site web statique pour votre compartiment. Au bas de la page, sous Static website hosting (Hébergement de site Web statique), vous voyez le point de terminaison du site web pour votre compartiment.

11. Sous Static website hosting (Hébergement de site Web statique), notez la valeur de Endpoint (Point de terminaison).

    Endpoint (Point de terminaison) correspond au point de terminaison du site web Amazon S3 de votre compartiment. Une fois que vous avez terminé de configurer votre compartiment en tant que site Web statique, vous pouvez utiliser ce point de terminaison pour tester votre site Web.

Utilisation de l'API REST

Pour plus d'informations sur l'envoi direct de demandes REST afin d'activer l'hébergement de site Web statique, consultez les sections suivantes dans le manuel Référence d'API Amazon Simple Storage Service :

- PUT Bucket website
- GET Bucket website
- DELETE Bucket website

Utilisation des AWS kits de développement logiciel

Pour héberger un site web statique dans Amazon S3, vous configurez un compartiment Amazon S3 pour l'hébergement de site web, puis vous chargez le contenu du site web dans le compartiment. Vous pouvez également utiliser les kits SDK AWS pour créer, mettre à jour et supprimer la configuration du site web par programmation. Les kits SDK fournissent des classes de type Wrapper autour de l'API REST Amazon S3. Vous pouvez envoyer des demandes d'API REST directement à partir de l'application, si cette dernière l'exige.

.NET

L'exemple suivant montre comment utiliser le AWS SDK for .NET pour gérer la configuration du site Web d'un bucket. Pour ajouter une configuration de site Web à un compartiment, vous fournissez un nom de compartiment et une configuration de site Web. La configuration de site Web doit comprendre un document d'index et peut contenir un rapport d'erreur facultatif. Ces documents doivent être stockés dans le compartiment. Pour plus d'informations, consultez PUT Bucket website. Pour plus d'informations sur la fonctionnalité du site Web Amazon S3, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.

L'exemple de code C# suivant ajoute une configuration de site Web sur le compartiment spécifié. La configuration spécifie les noms du document d'index et du rapport d'erreur. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string indexDocumentSuffix = "*** index object key ***"; //
 For example, index.html.
        private const string errorDocument = "*** error object key ***"; // For
 example, error.html.
        // Specify your bucket region (an example region is shown).
```

```
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
errorDocument).Wait();
        }

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                                            string indexDocumentSuffix,
                                            string errorDocument)
        {
            try
            {
                // 1. Put the website configuration.
                PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument
                    }
                };
                PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);

                // 2. Get the website configuration.
                GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
                {
                    BucketName = bucketName
                };
                GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
                Console.WriteLine("Index document: {0}",
getResponse.WebsiteConfiguration.IndexDocumentSuffix);
                Console.WriteLine("Error document: {0}",
getResponse.WebsiteConfiguration.ErrorDocument);
            }
            catch (AmazonS3Exception e)
            {
```

```
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

L'exemple PHP suivant ajoute une configuration de site Web au compartiment spécifié. La méthode `create_website_config` fournit explicitement les noms du document d'index et du rapport d'erreur. L'exemple récupère également la configuration de site Web et affiche la réponse. Pour de plus amples informations sur la fonction Amazon S3 website, veuillez consulter Hébergement d'un site Web statique à l'aide d'Amazon S3.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);


// Add the website configuration.
$s3->putBucketWebsite([
    'Bucket'                => $bucket,
    'WebsiteConfiguration'  => [
        'IndexDocument' => ['Suffix' => 'index.html'],
        'ErrorDocument' => ['Key' => 'error.html']
    ]
]);
```

```
// Retrieve the website configuration.
$result = $s3->getBucketWebsite([
    'Bucket' => $bucket
]);
echo $result->getPath('IndexDocument/Suffix');

// Delete the website configuration.
$s3->deleteBucketWebsite([
    'Bucket' => $bucket
]);
```

À l'aide du AWS CLI

Pour plus d'informations sur l'utilisation du AWS CLI pour configurer un compartiment S3 en tant que site Web statique, consultez le site Web dans le manuel de référence des AWS CLI commandes.

Ensuite, vous devez configurer le document d'index et définir les autorisations. Pour plus d'informations, consultez Configuration d'un document d'index et Définition des autorisations pour l'accès au site web.

Vous pouvez également configurer un document d'erreur, la journalisation du trafic web ou une redirection.

Configuration d'un document d'index

Lorsque vous activez l'hébergement de site web, vous devez également configurer et charger un document d'index. Un document d'index est une page web renvoyée par Amazon S3 quand une demande est effectuée au niveau de la racine d'un site web ou de n'importe quel sous-dossier. A titre d'exemple, si un utilisateur entre `http://www.example.com` dans le navigateur, l'utilisateur ne demande pas de page spécifique. Dans ce cas, Amazon S3 affiche le document d'index, qui est parfois appelé page par défaut.

Lorsque vous activez l'hébergement de site web statique pour votre compartiment, vous saisissez le nom du document d'index (par exemple, `index.html`). Après avoir activé l'hébergement de site web statique pour votre compartiment, vous chargez un fichier HTML avec le nom du document d'index dans votre compartiment.

La barre oblique à la fin de l'URL racine est facultative. A titre d'exemple, si vous configurez votre site web avec `index.html` comme document d'index, les URL suivantes renvoient `index.html`.

```
http://example-bucket.s3-website.Region.amazonaws.com/
http://example-bucket.s3-website.Region.amazonaws.com
```

Pour plus d'informations sur les points de terminaison des sites web Amazon S3, consultez Points de terminaison de sites web.

Document d'index et dossiers

Dans Amazon S3, un compartiment est un conteneur plat d'objets. Il ne fournit pas une structure hiérarchique contrairement au système de fichiers sur votre ordinateur. Cependant, vous pouvez créer une hiérarchie logique à l'aide des noms de clés d'objet qui nécessitent une structure de dossiers.

Imaginons, par exemple, un compartiment avec trois objets et les noms de clés suivants : Bien qu'ils soient stockés sans structure hiérarchique physique, vous pouvez deviner l'arborescence logique des dossiers à partir des noms de clés :

- `sample1.jpg` - L'objet est situé à la racine du compartiment.
- `photos/2006/Jan/sample2.jpg` - L'objet est situé dans le sous-dossier `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg` - L'objet est situé dans le sous-dossier `photos/2006/Feb`.

Dans la console Amazon S3, vous pouvez également créer un dossier dans un compartiment. Par exemple, vous pouvez créer un dossier nommé `photos`. Vous pouvez charger des objets dans le compartiment ou dans le dossier `photos` du compartiment. Si vous ajoutez l'objet `sample.jpg` dans le compartiment, le nom de la clé est `sample.jpg`. Si vous chargez l'objet dans le dossier `photos`, le nom de la clé d'objet est `photos/sample.jpg`.

Si vous créez une structure de dossiers dans votre compartiment, vous devez avoir un document d'index à chaque niveau. Le document d'index doit avoir le même nom dans chaque dossier, par exemple, `index.html`. Lorsqu'un utilisateur spécifie une URL qui se présente comme une recherche de répertoire, la présence ou l'absence d'une barre oblique finale détermine le comportement du site Web. A titre d'exemple, l'URL suivante, avec une barre oblique finale, renvoie le document d'index `photos/index.html`.

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Néanmoins, si vous supprimez la barre oblique finale dans la précédente URL, Amazon S3 recherche d'abord un objet `photos` dans le compartiment. Si l'objet `photos` est introuvable, il cherche alors

un document d'index, `photos/index.html`. Si ce document est trouvé, Amazon S3 renvoie un message `302 Found` et pointe vers la clé `photos/`. Pour les demandes suivantes vers `photos/`, Amazon S3 renvoie `photos/index.html`. Si le document d'index est introuvable, Amazon S3 renvoie une erreur.

Configurer un document d'index

Pour configurer un document d'index à l'aide de la console S3, procédez comme suit. Vous pouvez également configurer un document d'index à l'aide de l'API REST, AWS des SDK AWS CLI, du ou AWS CloudFormation.

> **Note**
>
> Dans un compartiment activé pour la gestion des versions, vous pouvez charger plusieurs copies de `index.html`, mais seule la version la plus récente est traitée. Pour plus d'informations sur la gestion des versions S3, consultez Utilisation de la gestion des versions dans les compartiments S3.

Lorsque vous activez l'hébergement de site web statique pour votre compartiment, vous saisissez le nom du document d'index (par exemple, **`index.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous téléchargez un fichier HTML avec le nom du document de cet index dans votre compartiment.

Pour configurer le document d'index

1. Créez un fichier `index.html`.

   Si vous n'avez pas de ficher `index.html`, vous pouvez utiliser le code HTML suivant pour en créer un :

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" >
   <head>
       <title>My Website Home Page</title>
   </head>
   <body>
     <h1>Welcome to my website</h1>
     <p>Now hosted on Amazon S3!</p>
   </body>
   </html>
   ```

2. Enregistrez le fichier d'index au niveau local.

   Le nom du fichier du document d'index doit correspondre exactement au nom du document d'index que vous saisissez dans la boîte de dialogue Hébergement de site Web statique . Le nom du document d'index est sensible à la casse. Par exemple, si vous saisissez `index.html` pour le nom du Document d'index dans la boîte de dialogue Hébergement de site Web statique, le nom du fichier de votre document d'index doit également être `index.html` et non `Index.html`.

3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.

5. Activez l'hébergement de site Web statique pour votre compartiment et saisissez le nom exact de votre document d'index (par exemple, `index.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.

6. Pour charger le document d'index dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier d'index dans la liste du compartiment de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

7. (Facultatif) Chargez du contenus d'un autre site Web dans votre compartiment.

Ensuite, vous devez définir des autorisations pour l'accès au site web. Pour plus d'informations, consultez Définition des autorisations pour l'accès au site web.

Vous pouvez également configurer un document d'erreur, la journalisation du trafic web ou une redirection.

Configuration d'un document d'erreur personnalisé

Une fois que vous avez configuré votre compartiment en tant que site web statique, lorsqu'une erreur se produit, Amazon S3 renvoie un document d'erreur HTML. Vous pouvez éventuellement configurer votre compartiment avec un document d'erreur personnalisé afin que Amazon S3 renvoie ce document en cas d'erreur.

> **Note**
>
> Certains navigateurs affichent leur propre message d'erreur lorsqu'une erreur se produit, sans tenir compte du document d'erreur renvoyé par Amazon S3. A titre d'exemple, lorsqu'une erreur HTTP 404 Not Found se produit, Google Chrome peut afficher son propre message d'erreur, sans tenir compte du document d'erreur renvoyé par Amazon S3.

Rubriques

Codes de réponse HTTP Amazon S3

Le tableau suivant répertorie le sous-ensemble des codes de réponse HTTP qu'Amazon S3 renvoie quand une erreur se produit.

Code d'erreur HTTP	Description
301 – Déplacé de façon permanente	Lorsqu'un utilisateur envoie une demande directement au point de terminaison du site web Amazon S3 (`http://s3-website.` *`Region`*`.amazonaws.com/`), Amazon S3 renvoie une réponse 301 Moved Permanently (Déplacé de façon permanente) et redirige ces demandes vers `https://aws.amazon.com/s3/` .
302 – Trouvé	Quand Amazon S3 reçoit une demande pour une clé `x`, `http://`*`bucket-name`*`.s3-website.` *`Region`*`.amazonaws.com/x` , sans barre oblique finale, il recherche d'abord l'objet portant le nom de clé `x`. Si l'objet est

Code d'erreur HTTP	Description
	introuvable, Amazon S3 en conclut que la demande concerne le sous-doss ier x et redirige la demande en ajoutant une barre oblique à la fin, puis renvoie le code 302 Found (Trouvé).
304 – Non modifié	Amazon S3 utilise des en-têtes de requête `If-Modified-Since` , `If-Unmodified-Since` , `If-Match` et/ou `If-None-Match` pour déterminer si l'objet demandé est identique à la copie mise en cache conservée par le client. Si l'objet est le même, le point de terminaison de site web renvoie une réponse 304 – Non modifié.
400 – Demande incorrecte	Le point de terminaison de site web renvoie un code 400 – Demande incorrecte quand l'utilisateur tente d'accéder à un compartiment via le point de terminaison Régional incorrect.
403 – Interdit	Le point de terminaison de site web renvoie le code d'erreur 403 – Interdit quand la demande de l'utilisateur est traduite en objet qui n'est pas accessibl e en lecture au public. Le propriétaire de l'objet doit permettre l'accès à l'objet en lecture au public via la stratégie de compartiment ou une liste ACL.

Code d'erreur HTTP	Description
404 – Non trouvé	Le point de terminaison de site web renvoie le code d'erreur 404 – Non trouvé pour les raisons suivantes : • Amazon S3 identifie que l'URL du site web fait référence à une clé d'objet qui n'existe pas. • Amazon S3 en déduit que la demande concerne un document d'index qui n'existe pas. • Le compartiment spécifié dans l'URL n'existe pas. • Le compartiment spécifié dans l'URL existe, mais n'est pas configuré comme un site web. Vous pouvez créer un document personnalisé qui sera renvoyé dans le cadre du code 404 – Non trouvé. Assurez-vous que le document est chargé dans le compartiment configuré comme un site web et que la configuration d'hébergement du site web est définie pour utiliser le document. Pour plus d'informations sur la façon dont Amazon S3 interprète l'URL comme une demande pour un objet ou un document d'index, consultez Configuration d'un document d'index.
500 – Erreur service	Le point de terminaison de site web répond avec le code 500 – Erreur service quand une erreur interne du serveur se produit.
503 – Service non disponible	Le point de terminaison du site web répond avec le code 503 Service Unavailable (Service non disponible) quand Amazon S3 détermine que vous devez réduire le débit des demandes.

Pour chacune de ces erreurs, Amazon S3 renvoie un message HTML prédéfini. Voici un exemple de message HTML renvoyé pour une réponse 403 – Accès interdit.

Configuration d'un document d'erreur personnalisé

Lorsque vous configurez votre compartiment en tant que site Web statique, vous pouvez fournir un document d'erreur personnalisé contenant un message d'erreur explicite et des conseils supplémentaires. Amazon S3 renvoie votre document d'erreur personnalisé uniquement pour la classe HTTP 4xx des codes d'erreur.

Pour configurer un document d'erreur personnalisé à l'aide de la console S3, procédez comme suit. Vous pouvez également configurer un document d'erreur à l'aide de l'API REST, AWS des SDK AWS CLI, du ou AWS CloudFormation. Pour plus d'informations, consultez les ressources suivantes :

- PutBucketWebsitedans le manuel de référence de l'API Amazon Simple Storage Service
- AWS::S3::Bucket WebsiteConfiguration dans le guide de l'utilisateur AWS CloudFormation
- put-bucket-website dans la référence de commande de l'AWS CLI

Lorsque vous activez l'hébergement de site Web statique pour votre compartiment, vous entrez le nom du document d'erreur (par exemple, **`404.html`**). Après avoir activé l'hébergement de site web statique pour le compartiment, vous chargez un fichier HTML avec le nom du document d'erreur dans votre compartiment.

Pour configurer un document d'erreur

1. Créez un document d'erreur, par exemple `404.html`.

2. Enregistrez le fichier de document d'erreur au niveau local.

   Le nom du document d'erreur est sensible à la casse et doit correspondre exactement au nom que vous saisissez lorsque vous activez l'hébergement de site web statique. Par exemple, si vous entrez `404.html` pour le nom du document d'Erreur dans la boîte de dialogue Hébergement de site Web statique, le nom de fichier de votre document d'erreur doit également être `404.html`.

3. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.

4. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment que vous souhaitez utiliser pour héberger un site Web statique.

5. Activez l'hébergement de site Web statique pour votre compartiment et entrez le nom exact de votre document d'erreur (par exemple, `404.html`). Pour plus d'informations, consultez Activation de l'hébergement de sites web et Configuration d'un document d'erreur personnalisé.

   Après l'activation de l'hébergement de site web statique, passez à l'étape 6.

6. Pour charger le document d'erreur dans votre compartiment, effectuez l'une des opérations suivantes :

   - Faites glisser et déposez le fichier du document d'erreur dans la liste des compartiments de la console.
   - Choisissez Upload (Charger), puis suivez les instructions pour choisir et charger le fichier d'index.

   Pour step-by-step obtenir des instructions, voirChargement d'objets.

Définition des autorisations pour l'accès au site web

Lorsque vous configurez un compartiment en tant que site web statique, si vous voulez que votre site web soit public, vous pouvez accorder un accès public en lecture. Pour que votre compartiment soit lisible publiquement, vous devez désactiver les paramètres de blocage de l'accès public pour le compartiment et écrire une stratégie de compartiment qui octroie un accès en lecture public. Si votre compartiment contient des objets qui n'appartiennent pas au propriétaire du compartiment, il se peut que vous ayez également besoin d'ajouter une liste ACL des objets qui accorde l'accès en lecture à tous.

Si vous ne souhaitez pas désactiver les paramètres de blocage de l'accès public pour votre compartiment, mais que vous souhaitez tout de même que votre site Web soit public, vous pouvez créer une CloudFront distribution Amazon pour servir votre site Web statique. Pour plus d'informations, consultez Accélérez votre site Web avec Amazon CloudFront ou utilisez une CloudFront distribution Amazon pour servir un site Web statique dans le guide du développeur Amazon Route 53.

> **Note**
>
> Sur le point de terminaison du site web, si un utilisateur demande un objet qui n'existe pas, Amazon S3 renvoie le code réponse HTTP `404 (Not Found)`. Si l'objet existe, mais que vous n'avez pas l'autorisation en lecture dessus, le point de terminaison du site Web renvoie un code réponse HTTP `403 (Access Denied)`. L'utilisateur peut utiliser le code réponse pour déduire si un objet spécifique existe ou non. Si vous ne voulez pas ce type de comportement, vous ne devez pas activer la prise en charge du site Web pour votre compartiment.

Rubriques

Étape 1 : Modifier les paramètres de blocage de l'accès public S3

Si vous voulez configurer un compartiment existant en tant que site web statique ayant un accès public, vous devez modifier les paramètres de blocage de l'accès public pour ce compartiment. Vous devrez peut-être également modifier les paramètres de blocage de l'accès public au niveau de votre compte. Amazon S3 applique la combinaison la plus restrictive de paramètres de blocage de l'accès public au niveau du compartiment et du compte.

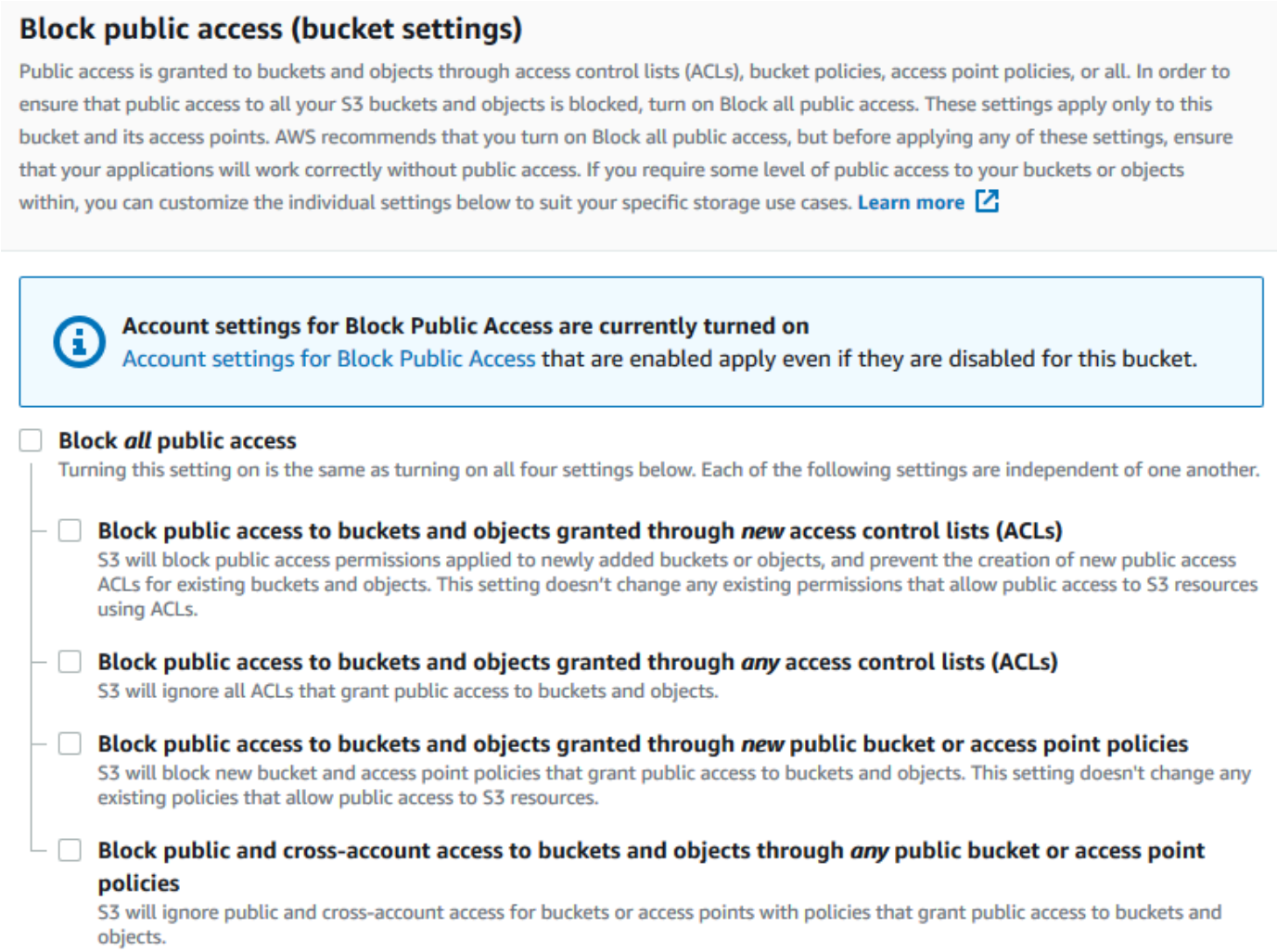
Par exemple, si vous autorisez l'accès public pour un compartiment, mais que vous bloquez tout accès public au niveau du compte, Amazon S3 continue de bloquer l'accès public au compartiment. Dans ce scénario, vous devez modifier vos paramètres de blocage de l'accès public de niveau compartiment et de niveau compte. Pour plus d'informations, consultez Blocage de l'accès public à votre stockage Amazon S3.

Par défaut, Amazon S3 bloque l'accès public à votre compte et à vos compartiments. Si vous souhaitez utiliser un compartiment pour héberger un site web statique, vous pouvez utiliser ces étapes pour modifier vos paramètres de blocage de l'accès public.

> **Warning**
>
> Avant de terminer cette étape, revoyez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tout accès public à vos compartiments.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Choisissez le nom du compartiment que vous avez configuré en tant que site web statique.
3. Choisissez Permissions.
4. Sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)), choisissez Edit (Modifier).
5. Effacez Block all public access (Bloquer tous les accès publics) et choisissez Enregistrer les modifications.

> **Warning**
>
> Avant de terminer cette étape, examinez Blocage de l'accès public à votre stockage Amazon S3 pour vous assurer que vous comprenez et acceptez les risques liés à l'autorisation d'accès public. Lorsque vous désactivez les paramètres de blocage de l'accès public pour rendre votre compartiment public, toute personne sur Internet peut accéder à votre compartiment. Nous vous recommandons de bloquer tous les accès publics à vos compartiments.



Amazon S3 désactive les paramètres de blocage de l'accès public pour votre compartiment. Pour créer un site web public statique, vous devrez peut-être aussi modifier les paramètres de blocage de l'accès public de votre compte avant d'ajouter une stratégie de compartiment. Si les paramètres du compte pour la fonctionnalité de blocage de l'accès public sont actuellement activés, une note s'affiche sous Block public access (bucket settings) (Bloquer l'accès public (paramètres de compartiment)).

Étape 2 : Ajouter une stratégie de compartiment

Pour que les objets de votre compartiment soient publiquement lisibles, vous devez écrire une stratégie de compartiment qui accorde à tous l'autorisation `s3:GetObject`.

Après avoir modifié les paramètres de blocage de l'accès public S3, vous devez ajouter une stratégie de compartiment pour accorder un accès public en lecture à votre compartiment. Lorsque vous accordez un accès public en lecture, tout le monde sur Internet peut accéder à votre compartiment.

> **Important**
>
> La stratégie suivante est uniquement un exemple et autorise un accès complet au contenu de votre compartiment. Avant d'effectuer cette étape, veuillez consulter Comment assurer la sécurité des fichiers de mon compartiment Amazon S3 ?, pour vous assurer que vous comprenez les bonnes pratiques pour sécuriser les fichiers dans votre compartiment S3 et les risques liés à l'octroi d'un accès public.

1. Dans Compartiments, choisissez le nom de votre compartiment.
2. Choisissez Permissions.
3. Sous Bucket Policy (Stratégie de compartiment), choisissez Edit (Modifier).
4. Pour accorder l'accès public en lecture à votre site web, copiez la stratégie de compartiment suivante et collez-la dans l'Éditeur de stratégie de compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Mettez à jour `Resource` pour inclure le nom de votre compartiment.

   Dans l'exemple précédent de stratégie de compartiment, *`Bucket-Name`* est un espace réservé pour le nom du compartiment. Pour utiliser cette stratégie de compartiment avec votre

propre compartiment, vous devez mettre à jour ce nom pour qu'il corresponde à celui de votre compartiment.

6. Choisissez Enregistrer les modifications.

   Un message s'affiche indiquant que la stratégie de compartiment a été ajoutée avec succès.

   Si une erreur indique `Policy has invalid resource`, confirmez que le nom du compartiment dans la stratégie de compartiment correspond au nom de votre compartiment. Pour plus d'informations sur l'ajout d'une politique de compartiment, consultez Comment ajouter une politique de compartiment S3 ?

   Si vous recevez un message d'erreur et que vous ne pouvez pas enregistrer la stratégie de compartiment, vérifiez les paramètres de blocage de l'accès public de votre compte et de votre compartiment pour confirmer que vous autorisez l'accès public au compartiment.

Listes de contrôle d'accès à l'objet

Vous pouvez utiliser une stratégie de compartiment pour accorder des autorisations à vos objets. Cependant, la stratégie de compartiment s'applique uniquement aux objets appartenant au propriétaire du compartiment. Si votre compartiment contient des objets qui n'appartiennent pas au propriétaire du compartiment, l'autorisation READ publique sur ces objets doit être accordée à l'aide de la liste ACL des objets.

La propriété d'objets S3 est un paramètre Amazon S3 au niveau des compartiments que vous pouvez utiliser pour contrôler la propriété des objets qui sont chargés dans votre compartiment, ainsi que pour désactiver ou activer les listes ACL. Par défaut, la propriété des objets est définie sur le paramètre Propriétaire du compartiment appliqué et toutes les listes ACL sont désactivées. Lorsque les listes ACL sont désactivées, le propriétaire du compartiment détient tous les objets du compartiment et gère leur accès exclusivement au moyen de politiques de gestion des accès.

La majorité des cas d'utilisation modernes dans Amazon S3 ne nécessitent plus l'utilisation des listes ACL. Nous vous recommandons de maintenir les listes ACL désactivées, sauf dans des circonstances inhabituelles où vous devez contrôler l'accès individuellement pour chaque objet. Lorsque les listes ACL sont désactivées, vous pouvez utiliser des politiques pour contrôler l'accès à tous les objets de votre compartiment, quelle que soit la personne qui les a chargés dans votre compartiment. Pour plus d'informations, consultez Consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.

> **Important**
>
> Si votre compartiment utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3, vous devez utiliser des politiques pour accorder l'accès à votre compartiment et aux objets qu'il contient. Quand le paramètre Propriétaire du compartiment appliqué est activé, les demandes de définition des listes de contrôle d'accès (ACL) ou des listes ACL de mise à jour échouent et renvoient le code d'erreur `AccessControlListNotSupported`. Les demandes de lecture de listes ACL sont toujours prises en charge.

Pour qu'un objet soit accessible en lecture publiquement à l'aide d'une liste ACL, accordez une autorisation READ au groupe `AllUsers`, comme indiqué dans l'élément « grant » suivant. Ajoutez cet élément « grant » à la liste ACL d'objet. Pour plus d'informations sur la gestion des listes ACL, consultez la section Présentation de la liste de contrôle d'accès (ACL).

```
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
```

(Facultatif) Journalisation du trafic web

Vous pouvez éventuellement activer la journalisation des accès au serveur Amazon S3 pour un compartiment configuré en tant que site web statique. La journalisation des accès au serveur fournit des enregistrements détaillés pour les demandes soumises à votre compartiment. Pour plus d'informations, consultez Enregistrement de demandes avec journalisation des accès au serveur. Si vous envisagez d'utiliser Amazon CloudFront pour accélérer votre site Web, vous pouvez également utiliser la CloudFront journalisation. Pour plus d'informations, consultez la section Configuration et utilisation des journaux d'accès dans le manuel Amazon CloudFront Developer Guide.

Pour activer la journalisation des accès au serveur pour votre compartiment de site web statique

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

2. Dans la Région où vous avez créé le compartiment configuré en tant que site Web statique, créez un compartiment pour la journalisation, par exemple `logs.example.com`.
3. Créez un dossier pour les fichiers de journalisation des accès au serveur (par exemple, `logs`).
4. (Facultatif) Si vous souhaitez l'utiliser CloudFront pour améliorer les performances de votre site Web, créez un dossier pour les fichiers CloudFront journaux (par exemple,`cdn`).

   Pour plus d'informations, consultez Accélérez votre site Web avec Amazon CloudFront.
5. Dans la liste Buckets (Compartiments), choisissez le nom de votre compartiment.
6. Choisissez Propriétés.
7. Sous Server access logging (Journalisation des accès au serveur), choisissez Edit (Modifier).
8. Sélectionnez Activer.
9. Sous Target bucket (Compartiment cible), choisissez la destination du compartiment et du dossier pour les journaux d'accès au serveur :

   - Accédez à l'emplacement du dossier et du compartiment :
     1. Choisissez Browse S3 (Parcourir S3).
     2. Choisissez le nom du compartiment, puis le dossier des journaux.
     3. Choisissez Choose path (Sélectionnez un chemin).
   - Saisissez le chemin du compartiment S3, par ex., **`s3://logs.example.com/logs/`**.
10. Choisissez Enregistrer les modifications.

    Dans votre compartiment de journaux, vous pouvez désormais accéder à vos journaux. Amazon S3 copie les journaux d'accès du site web dans votre compartiment de journaux toutes les deux heures.

(Facultatif) Configuration de la redirection de pages web

Si votre compartiment Amazon S3 est configuré pour l'hébergement de site web statique, vous pouvez configurer des redirections pour votre compartiment ou les objets qu'il contient. Vous disposez des options suivantes pour configurer des redirections.

Rubriques

- Rediriger les demandes pour le point de terminaison de site web de votre compartiment vers un autre compartiment ou domaine

- Configurer des règles de redirection pour utiliser des redirections conditionnelles avancées
- Rediriger les demandes pour un objet

Rediriger les demandes pour le point de terminaison de site web de votre compartiment vers un autre compartiment ou domaine

Vous pouvez rediriger toutes les demandes d'un point de terminaison de site web pour un compartiment vers un autre compartiment ou domaine. Si vous redirigez toutes les demandes, chaque demande adressée au point de terminaison de site web est redirigée vers le compartiment ou le domaine spécifié.

Par exemple, si votre domaine racine est `example.com`, et que vous souhaitez gérer des demandes pour `http://example.com` et `http://www.example.com`, vous devez créer deux compartiments nommés `example.com` et `www.example.com`. Ensuite, gérez le contenu dans le compartiment `example.com` et configurez l'autre compartiment `www.example.com` de manière à rediriger toutes les demandes vers le compartiment `example.com`. Pour plus d'informations, consultez les informations relatives à la configuration d'un site Web statique à l'aide d'un nom de domaine personnalisé.

Pour rediriger des demandes pour un point de terminaison de site Web de compartiment

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Sous Buckets (Compartiments), choisissez le nom du compartiment à partir duquel vous souhaitez rediriger les demandes (par exemple, `www.example.com`).
3. Choisissez Propriétés.
4. Sous Static website hosting (Hébergement de site web statique), choisissez Edit (Modifier).
5. Choisissez Redirect requests for an object (Rediriger les demandes pour un objet).
6. Dans la zone Host name (Nom d'hôte), entrez le point de terminaison du site web pour votre compartiment ou votre domaine personnalisé.

   Par exemple, si vous redirigez les demandes vers une adresse de domaine racine, entrez **`example.com`**.
7. Pour Protocol (Protocole), choisissez le protocole pour les demandes redirigées (none (aucun),http ou https).

   Si vous ne spécifiez pas de protocole, l'option par défaut est none (aucun).

8. Sélectionnez Enregistrer les modifications.

Configurer des règles de redirection pour utiliser des redirections conditionnelles avancées

Avec les règles de redirection avancées, vous pouvez acheminer des demandes de façon conditionnelle, en fonction de noms de clés d'objets spécifiques, de préfixes dans la demande ou de codes réponse. A titre d'exemple, supposez que vous supprimiez ou renommiez un objet dans votre compartiment. Vous pouvez ajouter une règle de routage qui redirige la demande vers un autre objet. Si vous souhaitez qu'un dossier ne soit plus accessible, vous pouvez ajouter une règle de routage pour rediriger la demande vers une autre page Web. Vous pouvez également ajouter une règle de routage pour gérer des conditions d'erreur en acheminant les demandes qui renvoient l'erreur vers un autre domaine, où celle-ci sera traitée.

Lorsque vous activez l'hébergement de site web statique pour votre compartiment, vous pouvez éventuellement spécifier des règles de redirection avancées. Amazon S3 impose une limitation de 50 règles de routage par configuration de site web. Si vous avez besoin de plus de 50 règles de routage, vous pouvez utiliser la redirection d'objet. Pour plus d'informations, consultez Utiliser la console S3..

Pour plus d'informations sur la configuration des règles de routage à l'aide de l'API REST, consultez PutBucketWebsitele manuel Amazon Simple Storage Service API Reference.

> **Important**
>
> Pour créer des règles de redirection dans la nouvelle console Amazon S3, vous devez utiliser JSON. Pour des exemples JSON, consultez Exemples de règles de redirection.

Pour configurer des règles de redirection pour un site web statique

Pour ajouter des règles de redirection pour un compartiment qui a déjà l'hébergement de site Web statique activé, procédez comme suit.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments, choisissez le nom d'un compartiment que vous avez configuré en tant que site web statique.

3. Choisissez Propriétés.
4. Sous Static website hosting (Hébergement de site Web statique), choisissez Edit (Modifier).
5. Dans la zone Redirection rules (Règles de redirection), entrez vos règles de redirection dans JSON.

   Dans la console S3, vous décrivez les règles en utilisant JSON. Pour des exemples JSON, consultez Exemples de règles de redirection. Amazon S3 impose une limitation de 50 règles de routage par configuration de site web.
6. Sélectionnez Enregistrer les modifications.

Éléments de règle de routage

Voici la syntaxe générale permettant de définir les règles de routage dans une configuration de site Web en JSON et XML Pour configurer les règles de redirection dans la nouvelle console S3, vous devez utiliser JSON. Pour des exemples JSON, consultez Exemples de règles de redirection.

JSON

```
[
    {
      "Condition": {
        "HttpErrorCodeReturnedEquals": "string",
        "KeyPrefixEquals": "string"
      },
      "Redirect": {
        "HostName": "string",
        "HttpRedirectCode": "string",
        "Protocol": "http"|"https",
        "ReplaceKeyPrefixWith": "string",
        "ReplaceKeyWith": "string"
      }
    }
  ]

Note: Redirect must each have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

XML

```
<RoutingRules> =
```

```
    <RoutingRules>
         <RoutingRule>...</RoutingRule>
         [<RoutingRule>...</RoutingRule>
         ...]
    </RoutingRules>

<RoutingRule> =
   <RoutingRule>
      [ <Condition>...</Condition> ]
      <Redirect>...</Redirect>
   </RoutingRule>

<Condition> =
   <Condition>
      [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
      [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
   </Condition>
    Note: <Condition> must have at least one child element.

<Redirect> =
   <Redirect>
      [ <HostName>...</HostName> ]
      [ <Protocol>...</Protocol> ]
      [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith>  ]
      [ <ReplaceKeyWith>...</ReplaceKeyWith> ]
      [ <HttpRedirectCode>...</HttpRedirectCode> ]
   </Redirect>

Note: <Redirect> must have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

Le tableau suivant décrit les éléments de la règle de routage.

Nom	Description
`RoutingRules`	Conteneur de l'ensemble des éléments `RoutingRule` .
`RoutingRule`	Une règle qui identifie une condition et la redirection qui est appliquée quand la condition est respectée. Condition :

Nom	Description
	• un conteneur `RoutingRules` doit comprendre au moins une règle de routage.
`Condition`	Conteneur de la description de la condition qui doit être respectée pour que la redirection spécifiée soit appliquée. Si la règle de routage ne comprend pas de condition, la règle s'applique à toutes les demandes.
`KeyPrefixEquals`	Le préfixe de nom de la clé d'objet à partir duquel les demandes sont redirigées. `KeyPrefixEquals` est requis si `HttpErrorCodeReturnedEquals` n'est pas spécifié. Si les deux paramètres `KeyPrefixEquals` et `HttpErrorCodeReturnedEquals` sont spécifiés, les deux doivent être vrais pour que la condition soit respectée.
`HttpErrorCodeReturnedEquals`	Le code d'erreur HTTP doit correspondre pour que la redirection s'applique. En cas d'erreur, et si le code d'erreur correspond à cette valeur, la redirection spécifiée est appliquée. `HttpErrorCodeReturnedEquals` est requis si `KeyPrefixEquals` n'est pas spécifié. Si les deux paramètres `KeyPrefixEquals` et `HttpErrorCodeReturnedEquals` sont spécifiés, les deux doivent être vrais pour que la condition soit respectée.

Nom	Description
`Redirect`	Elément du conteneur qui fournit des instructions pour rediriger la demande. Vous pouvez rediriger les demandes vers un autre hôte ou une autre page ; vous pouvez également indiquer un autre protocole d'utilisation. Une règle `RoutingRule` doit avoir un élément `Redirect`. Un élément `Redirect` doit contenir au moins l'un des éléments enfants suivants : `Protocol`, `HostName`, `ReplaceKeyPrefixWith` , `ReplaceKeyWith` ou `HttpRedirectCode` .
`Protocol`	Le protocole, `http` or `https`, à utiliser dans l'en-tête `Location` renvoyé dans la réponse. Si l'un des enfants est fourni, `Protocol` n'est pas requis.
`HostName`	Le nom d'hôte à utiliser dans l'en-tête `Location` qui est renvoyé dans la réponse. Si l'un des enfants est fourni, `HostName` n'est pas requis.
`ReplaceKeyPrefixWith`	Le préfixe de nom de la clé d'objet qui remplace la valeur de `KeyPrefixEquals` dans la demande de redirection. Si l'un des enfants est fourni, `ReplaceKeyPrefixWith` n'est pas requis. Il peut être fourni uniquement si `ReplaceKeyWith` n'est pas fourni.
`ReplaceKeyWith`	La clé de l'objet à utiliser dans l'en-tête `Location` qui est renvoyé dans la réponse. Si l'un des enfants est fourni, `ReplaceKeyWith` n'est pas requis. Il peut être fourni uniquement si `ReplaceKeyPrefixWith` n'est pas fourni.

Nom	Description
`HttpRedirectCode`	Le code de redirection HTTP à utiliser dans l'en-tête `Location` qui est renvoyé dans la réponse. Si l'un des enfants est fourni, `HttpRedirectCode` n'est pas requis.

Exemples de règles de redirection

Les exemples suivants décrivent les tâches de redirection courantes :

> **Important**
>
> Pour créer des règles de redirection dans la nouvelle console Amazon S3, vous devez utiliser JSON.

Example 1 : Redirection après renommage du préfixe de clé

Supposez que votre compartiment contienne les objets suivants :

- index.html
- docs/article1.html
- docs/article2.html

Vous décidez de remplacer le nom de dossier `docs/` par `documents/`. Après avoir fait ce changement, vous devez rediriger les demandes pour le préfixe `docs/` vers `documents/`. A titre d'exemple, la demande pour `docs/article1.html` sera redirigée vers `documents/article1.html`.

Dans ce cas, vous ajoutez la règle de routage suivante à la configuration du site web.

JSON

```
[
    {
```

```
        "Condition": {
            "KeyPrefixEquals": "docs/"
        },
        "Redirect": {
            "ReplaceKeyPrefixWith": "documents/"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Example 2 : Redirection des demandes pour un répertoire supprimé vers une page

Supposez que vous supprimiez le dossier `images/`(c'est-à-dire tous les objets ayant le préfixe de clé `images/`). Vous pouvez ajouter une règle de routage qui redirige les demandes concernant tout objet ayant le préfixe de clé `images/` vers une page appelée `folderdeleted.html`.

JSON

```
[
    {
        "Condition": {
            "KeyPrefixEquals": "images/"
        },
        "Redirect": {
            "ReplaceKeyWith": "folderdeleted.html"
        }
    }
]
```

XML

```
<RoutingRules>
  <RoutingRule>
  <Condition>
     <KeyPrefixEquals>images/</KeyPrefixEquals>
  </Condition>
  <Redirect>
    <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
  </Redirect>
  </RoutingRule>
</RoutingRules>
```

Example 3 : Redirection d'une erreur HTTP

Supposons que, lorsqu'un objet demandé est introuvable, vous souhaitiez rediriger les demandes vers une instance Amazon Elastic Compute Cloud (Amazon EC2). Ajoutez une règle de redirection pour que le visiteur du site soit redirigé vers une instance Amazon EC2 qui gère la demande dès qu'un code de statut HTTP 404 (Not Found (Non trouvé)) est renvoyé.

L'exemple suivant insère également le préfixe de la clé d'objet `report-404/` dans la redirection. À titre d'exemple, si vous demandez une page `ExamplePage.html` et qu'une erreur HTTP 404 est renvoyée, la demande est redirigée vers une page `report-404/ExamplePage.html` sur l'instance Amazon EC2 spécifiée. S'il n'y a pas de règle de routage et qu'une erreur HTTP 404 se produit, le document d'erreur spécifié dans la configuration est renvoyé.

JSON

```
[
    {
        "Condition": {
            "HttpErrorCodeReturnedEquals": "404"
        },
        "Redirect": {
            "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",
            "ReplaceKeyPrefixWith": "report-404/"
        }
    }
]
```

XML

```
<RoutingRules>
  <RoutingRule>
  <Condition>
    <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
  </Condition>
  <Redirect>
    <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
    <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
  </Redirect>
  </RoutingRule>
</RoutingRules>
```

Rediriger les demandes pour un objet

Vous pouvez rediriger les demandes d'un objet vers un autre objet ou une autre URL en définissant l'emplacement de redirection du site Web dans les métadonnées de l'objet. Vous configurez la redirection en ajoutant la propriété `x-amz-website-redirect-location` aux métadonnées d'objet. Sur la console Amazon S3, vous définissez l'élément Website Redirect Location (Emplacement de redirection de site web) dans les métadonnées de l'objet. Si vous utilisez l'API Amazon S3, vous définissez `x-amz-website-redirect-location`. Le site web interprète alors l'objet comme une redirection 301.

Pour rediriger une demande vers un autre objet, vous définissez l'emplacement de redirection vers la clé de l'objet cible. Pour rediriger une demande vers une URL externe, vous définissez l'emplacement de redirection vers l'URL choisie. Pour plus d'informations sur les métadonnées d'objet, consultez Métadonnées d'objet définies par le système.

Quand vous définissez une redirection de page, vous pouvez garder ou supprimer le contenu de l'objet source. Par exemple, si vous avez un objet `page1.html` dans votre compartiment, vous pouvez rediriger toutes les demandes de cette page vers un autre objet, `page2.html`. Vous avez deux options :

- Conservez le contenu de l'objet `page1.html` et redirigez les demandes de pages.
- Supprimez le contenu de `page1.html` et chargez un objet zéro octet nommé `page1.html` pour remplacer l'objet existant et rediriger les demandes de pages.

Utiliser la console S3.

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans la liste Buckets (Compartiments), choisissez le nom d'un compartiment que vous avez configuré en tant que site web statique (par exemple, `example.com`).
3. Sous Objects (Objets), sélectionnez votre objet.
4. Choisissez Actions, puis Edit metadata (Modifier les métadonnées).
5. Choisissez Métadonnées.
6. Choisissez Add Metadata (Ajouter des métadonnées).
7. Sous Type, choisissez System Defined (Défini par le système).
8. Dans Key, choisissez x-amz-website-redirect-location.
9. Dans Valeur, entrez le nom de clé de l'objet vers lequel vous souhaitez rediriger, par exemple `/page2.html`.

   Pour un autre objet dans le même compartiment, le préfixe `/` dans la valeur est requis. Vous pouvez également définir la valeur sur une URL externe, par exempl, `http://www.example.com`.
10. Choisissez Edit metadata (Modifier les métadonnées).

Utilisation de l'API REST

Les actions d'API Amazon S3 suivantes prennent en charge l'en-tête `x-amz-website-redirect-location` dans la demande. Amazon S3 stocke la valeur de l'en-tête dans les métadonnées de l'objet sous la forme `x-amz-website-redirect-location`.

- PUT Object
- Lancement du chargement partitionné
- POST Object
- PUT Object - Copy

Un compartiment configuré pour l'hébergement d'un site Web a un point de terminaison de site Web et un point de terminaison REST. Une demande pour une page qui est configurée comme une redirection 301 peut engendrer les résultats possibles suivants, selon le point de terminaison de la demande :

- Point de terminaison de site web propre à une Région – Amazon S3 redirige la demande de page en fonction de la valeur de la propriété `x-amz-website-redirect-location`.
- Point de terminaison REST – Amazon S3 ne redirige pas la demande de page. Il renvoie l'objet demandé.

Pour plus d'informations sur les points de terminaison, consultez Différences clés entre un point de terminaison de site web et un point de terminaison de l'API REST.

Lors de la définition d'une redirection de page, vous pouvez garder ou supprimer le contenu de l'objet source. A titre d'exemple, supposez que vous ayez un objet `page1.html` dans votre compartiment.

- Pour conserver le contenu de `page1.html` et rediriger uniquement les demandes de page, vous pouvez envoyer une demande PUT Object - Copy pour créer un objet `page1.html` qui utilise l'objet `page1.html` existant comme source. Dans votre demande, vous définissez l'en-tête `x-amz-website-redirect-location`. Lorsque la demande est terminée, la page d'origine et son contenu restent inchangés, cependant Amazon S3 redirige toute demande de la page vers l'emplacement de redirection que vous avez indiqué.
- Pour supprimer le contenu de l'objet `page1.html` et rediriger les demandes pour la page, vous pouvez envoyer une demande PUT Object pour charger un objet de 0 octet ayant la même clé d'objet : `page1.html`. Dans une demande PUT, vous définissez `x-amz-website-redirect-location` pour `page1.html`dans le nouvel objet. Une fois que la demande est terminée, `page1.html` n'a aucun contenu, et les demandes sont redirigées vers l'emplacement qui est spécifié par `x-amz-website-redirect-location`.

Lorsque vous récupérez l'objet à l'aide de l'action GET Object avec les autres métadonnées d'objet, Amazon S3 renvoie l'en-tête `x-amz-website-redirect-location` dans la réponse.

Développer avec Amazon S3

Cette section couvre les rubriques relatives aux développeurs concernant l'utilisation d'Amazon S3. Pour plus d'informations, consultez les rubriques ci-dessous.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

Demandes

Amazon S3 est un service REST. Vous pouvez envoyer des demandes à Amazon S3 en utilisant l'API REST ou les bibliothèques d'enveloppe du kit SDK AWS (consultez Exemples de code et de bibliothèques) qui enveloppent l'API REST Amazon S3 sous-jacente, simplifiant ainsi vos tâches de programmation.

Chaque interaction avec Amazon S3 est authentifiée ou anonyme. L'authentification est un processus de vérification de l'identité du demandeur essayant d'accéder à un produit Amazon Web Services (AWS). Les demandes authentifiées doivent inclure une valeur de signature qui authentifie l'expéditeur de la demande. La valeur de signature est en partie générée à partir des clés d'accès AWS du demandeur (ID de clé d'accès et clé d'accès secrète). Pour plus d'informations sur l'obtention des clés d'accès, consultez Comment puis-je obtenir les informations d'identification de sécurité ? dans le Références générales AWS.

Si vous utilisez le kit SDK AWS, les bibliothèques calculent la signature à partir des clés que vous fournissez. Cependant, si vous faites des appels directs d'API REST dans votre application, vous devez écrire le code pour calculer la signature et l'ajouter à la demande.

Rubriques

À propos des clés d'accès

Les sections suivantes examinent les types de clés d'accès que vous pouvez utiliser pour effectuer des demandes authentifiées.

Clés d'accès Compte AWS

Les clés d'accès au compte fournissent un accès complet aux ressources AWS détenues par le compte. Voici quelques exemples de clés d'accès :

- ID de clé d'accès (chaîne alphanumérique de 20 caractères). Par exemple : AKIAIOSFODNN7EXAMPLE
- Clé d'accès secrète (chaîne de 40 caractères). Par exemple : wjalrxutnfemi/k7mdeng/bpxrficyExampleKey

L'ID de clé d'accès identifie de manière unique un Compte AWS. Vous pouvez utiliser ces clés d'accès pour envoyer des demandes authentifiées à Amazon S3.

Clés d'accès utilisateur IAM

Vous pouvez créer un Compte AWS pour votre société. Cependant, plusieurs employés de l'organisation peuvent avoir besoin d'accéder aux ressources AWS de votre organisation. Le partage de vos clés d'accès de Compte AWS affaiblit la sécurité et la création de Comptes AWS individuels pour chaque employé peut ne pas être pratique. De plus, vous ne pouvez pas facilement partager des ressources comme les compartiments et les objets car elles appartiennent à différents comptes.

Pour partager des ressources, vous devez accorder des autorisations, ce qui représente du travail en plus.

Dans de tels scénarios, vous pouvez utiliser AWS Identity and Access Management (IAM) pour créer sous votre Compte AWS des utilisateurs avec leurs propres clés d'accès, et attacher des politiques d'utilisateur IAM qui accordent à ces utilisateurs des autorisations d'accès aux ressources appropriées. Pour mieux gérer ces utilisateurs, IAM vous permet de créer des groupes d'utilisateurs et d'accorder des autorisations au niveau du groupe qui s'appliquent à tous les utilisateurs de ce groupe.

Ces utilisateurs sont désignés comme les utilisateurs IAM que vous créez et gérez au sein d AWS. Le compte parent contrôle la possibilité de l'utilisateur à accéder à AWS. Toute ressource créée par un utilisateur IAM est sous le contrôle du Compte AWS parent et est payée par celui-ci. Ces utilisateurs IAM peuvent envoyer des demandes authentifiées à Amazon S3 en utilisant leurs propres informations d'identification de sécurité. Pour plus d'informations sur la création et la gestion des utilisateurs sous votre Compte AWS, accédez à la page des détails du produit AWS Identity and Access Management.

informations d'identification de sécurité temporaires

Outre la création d'utilisateurs IAM avec leurs propres clés d'accès, IAM vous permet d'octroyer des informations d'identification de sécurité temporaires (clés d'accès temporaires et un jeton de sécurité) à n'importe quel utilisateur IAM pour lui permettre d'accéder à vos services et ressources AWS. Vous pouvez aussi gérer des utilisateurs dans votre système hors d AWS. Ces dernières sont appelées utilisateurs fédérés. De plus, les utilisateurs peuvent être des applications que vous créez pour accéder à vos ressources AWS.

IAM fournit l'API AWS Security Token Service nécessaire pour demander des autorisations de sécurité temporaires. Vous pouvez utiliser l'API AWS STS ou le kit SDK AWS pour demander ces autorisations. L'API retourne des informations d'identification de sécurité temporaires (ID de clé d'accès et clé Secret Access Key), ainsi qu'un jeton de sécurité. Ces informations d'identification sont uniquement valides pour la durée que vous spécifiez lorsque vous les demandez. Vous utilisez l'ID de clé d'accès et la clé secrète de la même façon que vous les utilisez lorsque vous envoyez des demandes en utilisant votre Compte AWS ou les clés d'accès utilisateur IAM. De plus, vous devez inclure le jeton dans chaque demande que vous envoyez à Amazon S3.

Un utilisateur IAM peut demander ces informations d'identification de sécurité temporaires pour sa propre utilisation ou les transmettre à des utilisateurs fédérés ou à des applications. Lors d'une

demande d'informations d'identification de sécurité temporaires pour des utilisateurs fédérés, vous devez fournir un nom d'utilisateur et une stratégie IAM qui définit les autorisations que vous souhaitez associer à ces informations. L'utilisateur fédéré ne peut pas obtenir plus d'autorisations que l'utilisateur IAM parent qui a demandé les informations d'identification temporaires.

Vous pouvez utiliser ces informations d'identification de sécurité temporaires pour effectuer des demandes auprès d'Amazon S3. Les bibliothèques d'API calculent la valeur de signature nécessaire en utilisant ces informations d'identification pour authentifier votre demande. Si vous envoyez des demandes en utilisant des informations d'identification expirées, Amazon S3 les rejette.

Pour plus d'informations sur la signature de demandes en utilisant des informations d'identification de sécurité temporaires dans vos demandes d'API REST, consultez Signature et authentification des demandes REST. Pour des informations sur l'envoi de demandes à l'aide de kits SDK AWS, consultez Demandes à l'aide des kits SDK AWS.

Pour de plus amples informations sur la prise en charge des informations d'identification de sécurité temporaires par IAM, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM.

Pour plus de sécurité, vous pouvez demander une authentification multi-facteur (MFA) lors de l'accès à vos ressources Amazon S3 en configurant une stratégie de compartiment. Pour plus d'informations, consultez Exigence d'une MFA. Une fois que vous avez demandé une authentification multi-facteur (MFA) pour l'accès à vos ressources Amazon S3, la seule façon d'accéder à ces ressources est de fournir des informations d'identification temporaires créées avec une clé MFA. Pour de plus amples informations, veuillez consulter la page de détails Authentication multifacteur AWS et la section Configuration de l'accès aux API protégé par MFA du Guide de l'utilisateur IAM.

Points de terminaison de demande

Vous envoyez des demandes REST au point de terminaison prédéfini du service. Pour obtenir la liste de tous les services AWS et de leurs points de terminaison correspondants, consultez Régions et points de terminaison dans la Références générales AWS.

Envoi de demandes à Amazon S3 via IPv6

Amazon Simple Storage Service (Amazon S3) prend en charge l'accès aux compartiments S3 à l'aide du Protocole Internet version 6 (IPv6), en plus du protocole IPv4. Les points de terminaison à double pile Amazon S3 prennent en charge les demandes envoyées aux compartiments S3 via

IPv6 et IPv4. Il n'y a aucuns frais supplémentaires pour accéder à Amazon S3 via IPv6. Pour de plus amples informations sur la tarification, veuillez consulter Tarification Amazon S3.

Rubriques

Mise en route de l'envoi des demandes via IPv6

Pour envoyer une demande à un compartiment S3 via IPv6, vous devez utiliser un point de terminaison Dual-Stack. La section suivante décrit comment envoyer des demandes via IPv6 à l'aide de points de terminaison Dual-Stack.

Voici quelques éléments à connaître avant de tenter d'accéder à un compartiment via IPv6 :

- Le client et le réseau accédant au compartiment doivent être autorisés à utiliser le protocole IPv6.
- Les demandes de type hébergement virtuel et type chemin sont prises en charge pour un accès via IPv6. Pour plus d'informations, consultez Points de terminaison Amazon S3 Dual-Stack.
- Si vous utilisez le filtrage des adresses IP source dans vos politiques d'utilisateur ou de compartiment AWS Identity and Access Management (IAM), vous devez mettre à jour les politiques pour inclure les plages d'adresses IPv6. Pour plus d'informations, consultez Utilisation d'adresses IPv6 dans les stratégies IAM.
- Lorsque vous utilisez le protocole IPv6, les fichiers journaux d'accès au serveur génèrent les adresses IP au format IPv6. Vous devez mettre à jour les outils, scripts et logiciels existants utilisés pour analyser les fichiers journaux Amazon S3, de manière à ce qu'ils puissent également analyser les adresses `Remote IP` au format IPv6. Pour de plus amples informations, veuillez consulter Format des journaux d'accès au serveur Amazon S3 et Enregistrement de demandes avec journalisation des accès au serveur.

> **Note**
>
> Si vous rencontrez des problèmes liés à la présence d'adresses IPv6 dans les fichiers journaux, contactez AWS Support.

Envoi de demandes via IPv6 à l'aide de points de terminaison Dual-Stack

Pour envoyer des demandes à l'aide d'appels d'API Amazon S3 via IPv6, vous devez utiliser des points de terminaison Dual-Stack. Le fonctionnement des opérations d'API Amazon S3 est le même, que vous accédiez à Amazon S3 via IPv6 ou via IPv4. Les performances sont normalement également identiques.

Lorsque vous utilisez l'API REST, vous accédez directement à un point de terminaison Dual-Stack. Pour plus d'informations, consultez Points de terminaison Dual-Stack.

Lorsque vous utilisez le AWS Command Line Interface (AWS CLI) et AWS les SDK, vous pouvez utiliser un paramètre ou un indicateur pour passer à un point de terminaison à double pile. Vous pouvez également spécifier directement le point de terminaison Dual-Stack en remplaçant le point de terminaison Amazon S3 dans le fichier de configuration.

Un point de terminaison Dual-Stack peut être utilisé pour accéder à un compartiment via IPv6 à partir de chacun des éléments suivants :

- Le AWS CLI, tu voisEn utilisant des points de terminaison à double pile à partir du AWS CLI.
- Les AWS SDK, voirUtilisation de points de terminaison Dual-Stack avec les kits SDK AWS.
- L'API REST (consultez Envoi de demandes à des points de terminaison Dual-Stack à l'aide de l'API REST).

Fonctions non disponibles via IPv6

La fonctionnalité suivante n'est actuellement pas prise en charge lors de l'accès à un compartiment S3 via IPv6 : Hébergement de site web statique à partir d'un compartiment S3.

Utilisation d'adresses IPv6 dans les stratégies IAM

Avant de tenter d'accéder à un compartiment à l'aide du protocole IPv6, vous devez vérifier que toutes les stratégies d'utilisateur IAM ou de compartiment S3 utilisées pour le filtrage des adresses IP ont été mises à jour et comprennent bien les plages d'adresses IPv6. En cas d'absence de mise à jour des stratégies de filtrage des adresses IP pour gérer les adresses IPv6, les clients risquent de ne pas pouvoir accéder correctement au compartiment lorsqu'ils commenceront à utiliser le protocole IPv6. Pour de plus amples informations sur la gestion des autorisations d'accès avec IAM, veuillez consulter Identity and Access Management pour Amazon S3.

Les stratégies IAM qui permettent de filtrer les adresses IP utilisent des opérateurs de condition d'adresse IP. La stratégie de compartiment suivante identifie la plage d'adresses 54.240.143.* d'adresses IPv4 autorisées à l'aide d'opérateurs de condition d'adresse IP. Toute adresse IP se trouvant hors de cette plage se verra refuser l'accès au compartiment (`examplebucket`). Comme toutes les adresses IPv6 se trouvent hors de la plage autorisée, cette stratégie empêche les adresses IPv6 d'accéder à `examplebucket`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::examplebucket/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Vous pouvez modifier l'élément `Condition` de la stratégie de compartiment afin d'autoriser les plages d'adresses IPv4 (`54.240.143.0/24`) et IPv6 (`2001:DB8:1234:5678::/64`), comme illustré dans l'exemple suivant. Vous pouvez utiliser le même type de bloc `Condition` que celui indiqué dans l'exemple pour mettre à jour vos stratégies de compartiment et d'utilisateur IAM.

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
        }
```

Avant d'utiliser le protocole IPv6, vous devez mettre à jour toutes les stratégies de compartiment et d'utilisateur IAM pertinentes qui utilisent le filtrage des adresses IP afin d'autoriser les plages d'adresses IPv6. Nous vous recommandons de mettre à jour vos stratégies IAM avec les plages

d'adresses IPv6 de votre organisation, en plus des plages IPv4 existantes. Pour obtenir un exemple de stratégie de compartiment autorisant l'accès à la fois via IPv6 etIPv4, veuillez consulter Restriction de l'accès à des adresses IP spécifiques.

Vous pouvez consulter vos stratégies utilisateur IAM à l'aide de la console IAM à l'adresse https://console.aws.amazon.com/iam/. Pour de plus amples informations sur IAM, veuillez consulter le Guide de l'utilisateur IAM. Pour en savoir plus sur les stratégies de compartiment S3, consultez Ajout d'une stratégie de compartiment à l'aide de la console Amazon S3.

Test de compatibilité d'adresses IP

Si vous utilisez Linux/Unix ou Mac OS X, vous pouvez vérifier s'il vous est possible d'accéder à un point de terminaison Dual-Stack via IPv6 à l'aide de la commande `curl`, comme illustré dans l'exemple suivant :

Example

```
curl -v  http://s3.dualstack.us-west-2.amazonaws.com/
```

Les informations que vous obtenez doivent ressembler à celles de l'exemple ci-dessous. Si vous êtes connecté via IPv6, l'adresse IP connectée est une adresse IPv6.

```
* About to connect() to s3-us-west-2.amazonaws.com port 80 (#0)
*   Trying IPv6 address... connected
* Connected to s3.dualstack.us-west-2.amazonaws.com (IPv6 address) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1 OpenSSL/1.0.1t
 zlib/1.2.3
> Host: s3.dualstack.us-west-2.amazonaws.com
```

Si vous utilisez Microsoft Windows 7 ou Windows 10, vous pouvez vérifier s'il vous est possible d'accéder à un point de terminaison Dual-Stack via IPv6 ou IPv4 à l'aide de la commande `ping`, comme illustré dans l'exemple suivant.

```
ping ipv6.s3.dualstack.us-west-2.amazonaws.com
```

Utilisation des points de terminaison Dual-Stack Amazon S3

Les points de terminaison à double pile Amazon S3 prennent en charge les demandes envoyées aux compartiments S3 via IPv6 et IPv4. Cette section décrit comment utiliser les points de terminaison Dual-Stack (double pile).

Rubriques

Points de terminaison Amazon S3 Dual-Stack

Lorsque vous envoyez une demande à un point de terminaison Dual-Stack, l'URL du compartiment est résolue en une adresse IPv6 ou IPv4. Pour plus d'informations sur l'accès à un compartiment via IPv6, consultez Envoi de demandes à Amazon S3 via IPv6.

Lorsque vous utilisez l'API REST, vous accédez directement à un point de terminaison Amazon S3 à l'aide du nom de point de terminaison (URI). Vous pouvez accéder à un compartiment S3 via un point de terminaison Dual-Stack en utilisant un nom de point de terminaison de type hébergement virtuel ou chemin. Amazon S3 prend uniquement en charge les noms de points de terminaison Dual-Stack Régionaux ; vous devez donc spécifier la Région dans le nom.

Utilisez les conventions de dénomination suivantes pour les points de terminaison Dual-Stack de type hébergement virtuel ou chemin :

- Point de terminaison Dual-Stack de type hébergement virtuel :

  nom_compartiment.s3.dualstack.*Région_aws*.amazonaws.com

- Point de terminaison Dual-Stack de type chemin :

  s3.dualstack.*Région_aws*.amazonaws.com/*nom_compartiment*

Pour plus d'informations sur le style des noms de points de terminaison, consultez Accès à un compartiment Amazon S3 et liste des compartiments. Pour obtenir la liste des points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.

> **Important**
>
> Vous pouvez utiliser Transfer Acceleration avec les points de terminaison double pile (« Dual-Stack »). Pour plus d'informations, consultez Mise en route d'Amazon S3 Transfer Acceleration.

> **Note**
>
> Les deux types de points de terminaison de VPC permettant d'accéder à Amazon S3 (points de terminaison de VPC d'interface et points de terminaison de VPC de passerelle) ne prennent pas en charge le mode double pile. Pour plus d'informations sur les points de terminaison de VPC pour Amazon S3, consultez AWS PrivateLink pour Amazon S3.

Lorsque vous utilisez le AWS Command Line Interface (AWS CLI) et AWS les SDK, vous pouvez utiliser un paramètre ou un indicateur pour passer à un point de terminaison à double pile. Vous pouvez également spécifier directement le point de terminaison Dual-Stack en remplaçant le point de terminaison Amazon S3 dans le fichier de configuration. Les sections suivantes décrivent comment utiliser les points de terminaison à double pile à partir des SDK AWS CLI et des AWS SDK.

En utilisant des points de terminaison à double pile à partir du AWS CLI

Cette section fournit des exemples de AWS CLI commandes utilisées pour envoyer des demandes à un point de terminaison à double pile. Pour obtenir des instructions sur la configuration du AWS CLI, voirDéveloppement avec Amazon S3 à l'aide de la AWS CLI.

Vous définissez la valeur de configuration `use_dualstack_endpoint` sur `true` dans un profil de votre AWS Config fichier pour diriger toutes les demandes Amazon S3 effectuées par les `s3api` AWS CLI commandes `s3` and vers le point de terminaison à double pile pour la région spécifiée. Vous indiquez la Région dans le fichier de configuration ou dans une commande à l'aide de l'option `--region`.

Lorsque vous utilisez des points de terminaison à double pile avec les styles AWS CLI, `path` les deux sont `virtual` pris en charge. Le type d'adressage défini dans le fichier de configuration vérifie

que le nom du compartiment se trouve bien dans le nom d'hôte ou fait partie de l'URL. Par défaut, l'interface de ligne de commande tente, si possible, d'utiliser le style virtuel, mais revient au type chemin si nécessaire. Pour plus d'informations, consultez Configuration de la AWS CLI d'Amazon S3.

Vous pouvez également modifier la configuration à l'aide d'une commande, comme indiqué dans l'exemple ci-dessous, qui définit `use_dualstack_endpoint` sur `true` et `addressing_style` sur `virtual` dans le profil par défaut.

```
$ aws configure set default.s3.use_dualstack_endpoint true
$ aws configure set default.s3.addressing_style virtual
```

Si vous souhaitez utiliser un point de terminaison à double pile uniquement pour AWS CLI les commandes spécifiées (pas pour toutes les commandes), vous pouvez utiliser l'une des méthodes suivantes :

- Vous pouvez utiliser le point de terminaison Dual-Stack par commande en définissant le paramètre `--endpoint-url` sur `https://s3.dualstack.`*`aws-region`*`.amazonaws.com` ou sur `http://s3.dualstack.`*`aws-region`*`.amazonaws.com` pour toute commande `s3` ou `s3api`.

  ```
  $ aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
  ```

- Vous pouvez configurer des profils distincts dans votre AWS Config fichier. Par exemple, vous pouvez créer un profil qui définit `use_dualstack_endpoint` sur `true` et un profil qui ne définit pas `use_dualstack_endpoint`. Lorsque vous exécutez une commande, spécifiez le profil adéquat selon que vous comptez utiliser ou non le point de terminaison Dual-Stack.

> **Note**
>
> Lorsque vous utilisez le, AWS CLI vous ne pouvez actuellement pas utiliser l'accélération de transfert avec des points de terminaison à double pile. Cependant, le support pour AWS CLI le sera bientôt. Pour plus d'informations, consultez À l'aide du AWS CLI.

Utilisation de points de terminaison Dual-Stack avec les kits SDK AWS

Cette section fournit des exemples d'accès à un point de terminaison à double pile à l'aide des AWS SDK.

AWS SDK for Java exemple de point de terminaison à double pile

L'exemple suivant montre comment activer les points de terminaison Dual-Stack lors de la création d'un client Amazon S3 à l'aide du kit AWS SDK for Java.

Pour obtenir des instructions sur la création et le test d'un exemple Java fonctionnel, consultez Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;

public class DualStackEndpoints {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            // Create an Amazon S3 client with dual-stack endpoints enabled.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .withDualstackEnabled(true)
                    .build();

            s3Client.listObjects(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Si vous utilisez AWS SDK for Java le sous Windows, vous devrez peut-être définir la propriété de machine virtuelle Java (JVM) suivante :

```
java.net.preferIPv6Addresses=true
```

AWS Exemple de point de terminaison à double pile du SDK .NET

Lorsque vous utilisez le AWS SDK pour .NET, vous utilisez `AmazonS3Config` la classe pour activer l'utilisation d'un point de terminaison à double pile, comme indiqué dans l'exemple suivant.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DualStackEndpointTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            var config = new AmazonS3Config
            {
                UseDualstackEndpoint = true,
                RegionEndpoint = bucketRegion
            };
            client = new AmazonS3Client(config);
            Console.WriteLine("Listing objects stored in a bucket");
            ListingObjectsAsync().Wait();
        }

        private static async Task ListingObjectsAsync()
        {
            try
            {
                var request = new ListObjectsV2Request
                {
```

```
                    BucketName = bucketName,
                    MaxKeys = 10
                };
                ListObjectsV2Response response;
                do
                {
                    response = await client.ListObjectsV2Async(request);

                    // Process the response.
                    foreach (S3Object entry in response.S3Objects)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }
                    Console.WriteLine("Next Continuation Token: {0}",
 response.NextContinuationToken);
                    request.ContinuationToken = response.NextContinuationToken;
                } while (response.IsTruncated == true);
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Pour obtenir un exemple complet de listes d'objets avec .NET, consultez la section Liste des clés d'objet par programme.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

Utilisation de points de terminaison Dual-Stack avec l'API REST

Pour obtenir des informations sur la procédure permettant d'envoyer des demandes à des points de terminaison Dual-Stack à l'aide de l'API REST, consultez la section Envoi de demandes à des points de terminaison Dual-Stack à l'aide de l'API REST.

Demandes à l'aide des kits SDK AWS

Rubriques

Vous pouvez envoyer des demandes authentifiées à Amazon S3 en utilisant le kit SDK AWS ou en effectuant directement les appels d'API REST dans votre application. L'API du kit SDK AWS utilise les informations d'identification que vous fournissez pour calculer la signature pour l'authentification. Si vous utilisez l'API REST directement dans vos applications, vous devez écrire le code nécessaire pour calculer la signature d'authentification de votre demande. Pour obtenir la liste des kits SDK AWS disponibles, veuillez consulter Exemples de code et bibliothèques.

Faire des demandes à l'aide des Compte AWS informations d'identification utilisateur ou IAM

Vous pouvez utiliser vos informations d'identification de sécurité Compte AWS ou celles de l'utilisateur IAM pour envoyer des demandes authentifiées à Amazon S3. Cette section fournit des exemples de la manière dont vous pouvez envoyer des demandes authentifiées à l'aide du AWS SDK for Java AWS SDK for .NET, et AWS SDK for PHP. Pour obtenir la liste des AWS SDK disponibles, consultez la section Exemples de code et bibliothèques.

Chacun de ces AWS SDK utilise une chaîne de fournisseurs d'informations d'identification spécifique au SDK pour rechercher et utiliser les informations d'identification et effectuer des actions au nom du propriétaire des informations d'identification. Ce que toutes ces chaînes de fournisseurs d'informations d'identification ont en commun, c'est qu'elles recherchent toutes votre fichier AWS d'informations d'identification local.

Pour plus d'informations, consultez les rubriques ci-dessous :

Rubriques

Pour créer un fichier d' AWS informations d'identification local

Le moyen le plus simple de configurer les informations d'identification de vos AWS SDK consiste à utiliser un fichier d' AWS informations d'identification. Si vous utilisez le AWS Command Line Interface (AWS CLI), il se peut qu'un fichier d'informations d' AWS identification local soit déjà configuré. Sinon, suivez la procédure ci-dessous pour configurer en :

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Créez un utilisateur doté d'autorisations limitées aux services et actions auxquels votre code sera autorisé à accéder. Pour en savoir plus sur la création d'un utilisateur, veuillez consulter Création d'utilisateurs IAM (Console), puis suivez les instructions jusqu'à l'étape 8.
3. Choisissez Download .csv (Télécharger .csv) pour enregistrer une copie locale de vos informations d'identification AWS .
4. Sur votre ordinateur, créez le répertoire `.aws` dans le répertoire de base. Sur les systèmes Unix, par exemple Linux ou OS X, ce répertoire se trouve à l'emplacement suivant :

   ```
   ~/.aws
   ```

   Sur les systèmes Windows, il se trouve à l'emplacement suivant :

   ```
   %HOMEPATH%\.aws
   ```
5. Dans le répertoire `.aws`, créez un fichier appelé `credentials`.
6. Ouvrez le fichier `.csv` des informations d'identification que vous avez téléchargé à partir de la console IAM, puis copiez-en le contenu dans le fichier , en respectant le format suivant :

   ```
   [default]
   aws_access_key_id = your_access_key_id
   aws_secret_access_key = your_secret_access_key
   ```
7. Enregistrez le fichier `credentials`, puis supprimez le fichier `.csv` que vous avez téléchargé à l'étape 3.

Votre fichier d'informations d'identification partagé est désormais configuré sur votre ordinateur local, et il est prêt à être utilisé avec les AWS SDK.

Envoi de demandes authentifiées à l'aide des SDK AWS

Utilisez les AWS SDK pour envoyer des demandes authentifiées. Pour plus d'informations sur l'envoi de demandes authentifiées, consultez Informations d'identification de sécurité AWS ou Authentification dans IAM Identity Center.

Java

Pour envoyer des demandes authentifiées à Amazon S3 à l'aide de vos informations d'identification Compte AWS ou de celles de l'utilisateur IAM, procédez comme suit :

- Utilisez la classe `AmazonS3ClientBuilder` pour créer une instance `AmazonS3Client`.
- Exécutez l'une des méthodes `AmazonS3Client` pour envoyer des demandes à Amazon S3. Le client génère la signature nécessaire à partir des informations d'identification que vous fournissez et l'inclut dans la demande.

L'exemple suivant exécute les tâches précédentes. Pour plus d'informations sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsRequest;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.s3.model.S3ObjectSummary;

import java.io.IOException;
import java.util.List;

public class MakingRequests {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
```

```
        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Get a list of objects in the bucket, two at a time, and
            // print the name and size of each object.
            ListObjectsRequest listRequest = new
 ListObjectsRequest().withBucketName(bucketName).withMaxKeys(2);
            ObjectListing objects = s3Client.listObjects(listRequest);
            while (true) {
                List<S3ObjectSummary> summaries = objects.getObjectSummaries();
                for (S3ObjectSummary summary : summaries) {
                    System.out.printf("Object \"%s\" retrieved with size %d\n",
 summary.getKey(), summary.getSize());
                }
                if (objects.isTruncated()) {
                    objects = s3Client.listNextBatchOfObjects(objects);
                } else {
                    break;
                }
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Pour envoyer des demandes authentifiées à l'aide de vos informations d'identification Compte AWS ou de celles de l'utilisateur IAM, procédez comme suit :

- Créez une instance de la classe `AmazonS3Client`.

- Exécutez l'une des méthodes `AmazonS3Client` pour envoyer des demandes à Amazon S3. Le client génère la signature nécessaire à partir des informations d'identification fournies et l'inclut dans la demande envoyée à Amazon S3.

Pour de plus amples informations, veuillez consulter Faire des demandes à l'aide des Compte AWS informations d'identification utilisateur ou IAM.

> **Note**
>
> - Vous pouvez créer la classe `AmazonS3Client` sans fournir d'informations d'identification de sécurité. Les demandes envoyées via ce client sont anonymes et ne comportent pas de signature. Amazon S3 renvoie un message d'erreur si vous envoyez des demandes anonymes pour une ressource qui n'est pas disponible publiquement.
> - Vous pouvez créer un Compte AWS et créer les utilisateurs requis. Vous pouvez également gérer les informations d'identification pour ces utilisateurs. Vous avez besoin de ces informations d'identification pour exécuter la tâche dans l'exemple suivant. Pour de plus amples informations, veuillez consulter la section Configurer les informations d'identification AWS du Guide du développeur pour le kit AWS SDK for .NET .
>
>   Vous pouvez ensuite configurer votre application pour récupérer activement les profils et les informations d'identification, puis utiliser explicitement ces informations d'identification lors de la création d'un client AWS de service. Pour plus d'informations, consultez la section Accès aux informations d'identification et aux profils dans une application du Guide du développeur pour le kit AWS SDK for .NET .

L'exemple C# suivant montre comment exécuter les tâches précédentes. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class MakeS3RequestTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            using (client = new AmazonS3Client(bucketRegion))
            {
                Console.WriteLine("Listing objects stored in a bucket");
                ListingObjectsAsync().Wait();
            }
        }

        static async Task ListingObjectsAsync()
        {
            try
            {
                ListObjectsRequest request = new ListObjectsRequest
                {
                    BucketName = bucketName,
                    MaxKeys = 2
                };
                do
                {
                    ListObjectsResponse response = await
 client.ListObjectsAsync(request);
                    // Process the response.
                    foreach (S3Object entry in response.S3Objects)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If the response is truncated, set the marker to get the next
                    // set of keys.
                    if (response.IsTruncated)
                    {
                        request.Marker = response.NextMarker;
```

```
                    }
                    else
                    {
                        request = null;
                    }
                } while (request != null);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Pour obtenir des exemples pratiques, consultez Présentation des objets Amazon S3 et Présentation des compartiments. Vous pouvez tester ces exemples en utilisant vos informations d'identification Compte AWS ou celles d'un utilisateur IAM.

Par exemple, pour lister toutes les clés d'objet du compartiment, consultez Liste des clés d'objet par programme.

PHP

Cette section explique comment utiliser une classe de la version 3 pour envoyer des demandes authentifiées AWS SDK for PHP à l'aide de vos informations d'identification Compte AWS ou de celles de l'utilisateur IAM. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

L'exemple PHP suivant montre comment le client effectue une demande à l'aide de vos informations d'identification de sécurité pour répertorier tous les compartiments pour votre compte.

Example

```
require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
]);

// Retrieve the list of buckets.
$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    echo "Keys retrieved!" . PHP_EOL;

    // Print the list of objects to the page.
    foreach ($objects as $object) {
        echo $object['Key'] . PHP_EOL;
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Note

Vous pouvez créer la classe `S3Client` sans fournir d'informations d'identification de sécurité. Les demandes envoyées via ce client sont anonymes et ne comportent pas de signature. Amazon S3 renvoie un message d'erreur si vous envoyez des demandes anonymes pour une ressource qui n'est pas disponible publiquement. Pour plus d'informations, consultez Création de clients anonymes dans la documentation AWS SDK for PHP.

Pour obtenir des exemples pratiques, consultez Présentation des objets Amazon S3. Vous pouvez tester ces exemples à l'aide de vos informations d'identification Compte AWS ou de celles de l'utilisateur IAM.

Par obtenir un exemple de liste des clés d'objet dans un compartiment, consultez Liste des clés d'objet par programme.

Ruby

Avant de pouvoir utiliser la version 3 du AWS SDK for Ruby pour passer des appels à Amazon S3, vous devez définir les informations d' AWS accès que le SDK utilise pour vérifier votre accès à vos compartiments et objets. Si vous avez défini des informations d'identification partagées dans le profil AWS d'identification de votre système local, la version 3 du SDK pour Ruby peut utiliser ces informations d'identification sans que vous ayez à les déclarer dans votre code. Pour en savoir plus sur la définition d'informations d'identification partagées, consultez Faire des demandes à l'aide des Compte AWS informations d'identification utilisateur ou IAM.

L'extrait de code Ruby suivant utilise les informations d'identification contenues dans un fichier AWS d'informations d'identification partagé sur un ordinateur local pour authentifier une demande visant à obtenir tous les noms de clés d'objets d'un compartiment spécifique. Il exécute les opérations suivantes :

1. Crée une instance de la classe `Aws::S3::Client`.
2. Envoie une demande à Amazon S3 en énumérant les objets d'un compartiment à l'aide de la méthode `list_objects_v2` de `Aws::S3::Client`. Le client génère la valeur de signature nécessaire à partir des informations d' AWS identification figurant dans le fichier d'informations d'identification de votre ordinateur et l'inclut dans la demande qu'il envoie à Amazon S3.
3. Imprime le tableau des noms de clés d'objet vers le terminal.

Example

```
# Prerequisites:
#  - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
```

```
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'doc-example-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Example usage:
def run_me
  region = "us-west-2"
  bucket_name = "BUCKET_NAME"
  s3_client = Aws::S3::Client.new(region: region)

  exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Si vous ne disposez pas d'un fichier AWS d'informations d'identification local, vous pouvez toujours créer la `Aws::S3::Client` ressource et exécuter le code sur les buckets et les objets Amazon S3. Les demandes envoyées avec la version 3 du kit SDK pour Ruby sont anonymes et ne comportent aucune signature par défaut. Amazon S3 renvoie une erreur si vous envoyez des demandes anonymes pour une ressource qui n'est pas disponible publiquement.

Vous pouvez utiliser et développer l'extrait de code précédent pour les applications du kit SDK pour Ruby, comme illustré dans l'exemple suivant, plus complexe.

```
# Prerequisites:
#  - An existing Amazon S3 bucket.

require "aws-sdk-s3"

# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if all operations succeed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_bucket_objects?(s3_client, 'doc-example-bucket')
def list_bucket_objects?(s3_client, bucket_name)
  puts "Accessing the bucket named '#{bucket_name}'..."
  objects = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if objects.count.positive?
    puts "The object keys in this bucket are (first 50 objects):"
    objects.contents.each do |object|
      puts object.key
    end
  else
    puts "No objects found in this bucket."
  end

  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
  return false
end

# Example usage:
def run_me
  region = "us-west-2"
  bucket_name = "BUCKET_NAME"
  s3_client = Aws::S3::Client.new(region: region)
```

```
  exit 1 unless list_bucket_objects?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Go

Example

L'exemple suivant utilise les AWS informations d'identification chargées automatiquement par le SDK pour Go à partir du fichier d'informations d'identification partagé.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/s3"
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 s3Client := s3.NewFromConfig(sdkConfig)
 count := 10
 fmt.Printf("Let's list up to %v buckets for your account.\n", count)
 result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
 if err != nil {
  fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
  return
 }
```

```
 if len(result.Buckets) == 0 {
  fmt.Println("You don't have any buckets!")
 } else {
  if count > len(result.Buckets) {
   count = len(result.Buckets)
  }
  for _, bucket := range result.Buckets[:count] {
   fmt.Printf("\t%v\n", *bucket.Name)
  }
 }
}
```

Ressources connexes

- Développement avec Amazon S3 à l'aide des AWS SDK
- AWS SDK for PHP pour la classe Amazon S3 Aws\S3\S3Client
- Documentation AWS SDK for PHP

Envoi de demandes à l'aide des informations d'identification temporaires des utilisateurs IAM

Un utilisateur Compte AWS ou un utilisateur IAM peut demander des informations d'identification de sécurité temporaires et les utiliser pour envoyer des demandes authentifiées à Amazon S3. Cette section fournit des exemples d'utilisation des kits AWS SDK for Java SDK pour Java, .NET, and PHP afin d'obtenir des informations d'identification de sécurité temporaires et de les utiliser pour authentifier vos demandes à Amazon S3.

Java

Un utilisateur IAM ou un utilisateur Compte AWS peut demander des informations d'identification de sécurité temporaires (voirDemandes) en utilisant le AWS SDK for Java et les utiliser pour accéder à Amazon S3. Ces informations d'identification expirent à la fin de session spécifiée.

Par défaut, la session dure une heure. Si vous utilisez les informations d'identification des utilisateurs IAM, vous pouvez spécifier la durée (de 15 minutes à la durée maximale de session pour le rôle) lors de la demande d'informations d'identification de sécurité temporaires. Pour en savoir plus sur les informations d'identification de sécurité temporaires, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la création de demandes, consultez Demandes.

Pour obtenir des informations d'identification de sécurité temporaires et accéder à Amazon S3

1. Créez une instance de la classe `AWSSecurityTokenService`. Pour en savoir plus sur les informations d'identification, consultez Développement avec Amazon S3 à l'aide des AWS SDK.
2. Récupérez les informations d'identification de sécurité temporaires pour le rôle souhaité en appelant la méthode `assumeRole()` du client STS (Security Token Service).
3. Regroupez les informations d'identification de sécurité temporaires dans un objet `BasicSessionCredentials`. Vous utilisez cet objet pour fournir les informations d'identification de sécurité temporaires à votre client Amazon S3.
4. Créez une instance de la classe `AmazonS3Client` à l'aide des informations d'identification de sécurité temporaires. Vous envoyez des demandes à Amazon S3 grâce à ce client. Si vous envoyez des demandes à l'aide d'informations d'identification expirées, Amazon S3 renvoie une erreur.

> **Note**
>
> Si vous obtenez les informations d'identification de sécurité temporaires à l'aide des informations d'identification de sécurité du Compte AWS , les informations d'identification de sécurité temporaires ne sont valides que pendant une heure. Vous pouvez spécifier la durée de session uniquement si vous utilisez les informations d'identification des utilisateurs IAM pour demander une session.

L'exemple suivant répertorie un ensemble de clés d'objet dans le compartiment spécifié. L'exemple obtient les informations d'identification de sécurité temporaires pour une session et les utilise pour envoyer une demande authentifiée à Amazon S3.

Si vous souhaitez tester l'exemple à l'aide des informations d'identification de l'utilisateur IAM, vous devez créer un utilisateur IAM sous votre Compte AWS. Pour de plus amples informations sur la création d'un utilisateur IAM, veuillez consulter Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

Pour obtenir des instructions sur la création et le test d'un échantillon de travail, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.AssumeRoleResult;
import com.amazonaws.services.securitytoken.model.Credentials;

public class MakingRequestsWithIAMTempCredentials {
    public static void main(String[] args) {
        String clientRegion = "*** Client region ***";
        String roleARN = "*** ARN for role to be assumed ***";
        String roleSessionName = "*** Role session name ***";
```

```
        String bucketName = "*** Bucket name ***";

        try {
            // Creating the STS client is part of your trusted code. It has
            // the security credentials you use to obtain temporary security
credentials.
            AWSSecurityTokenService stsClient =
AWSSecurityTokenServiceClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Obtain credentials for the IAM role. Note that you cannot assume the
role of
            // an AWS root account;
            // Amazon S3 will deny access. You must use credentials for an IAM user
or an
            // IAM role.
            AssumeRoleRequest roleRequest = new AssumeRoleRequest()
                    .withRoleArn(roleARN)
                    .withRoleSessionName(roleSessionName);
            AssumeRoleResult roleResponse = stsClient.assumeRole(roleRequest);
            Credentials sessionCredentials = roleResponse.getCredentials();

            // Create a BasicSessionCredentials object that contains the credentials
you
            // just retrieved.
            BasicSessionCredentials awsCredentials = new BasicSessionCredentials(
                    sessionCredentials.getAccessKeyId(),
                    sessionCredentials.getSecretAccessKey(),
                    sessionCredentials.getSessionToken());

            // Provide temporary security credentials so that the Amazon S3 client
            // can send authenticated requests to Amazon S3. You create the client
            // using the sessionCredentials object.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new
AWSStaticCredentialsProvider(awsCredentials))
                    .withRegion(clientRegion)
                    .build();

            // Verify that assuming the role worked and the permissions are set
correctly
            // by getting a set of object keys from the bucket.
```

```
            ObjectListing objects = s3Client.listObjects(bucketName);
            System.out.println("No. of Objects: " +
 objects.getObjectSummaries().size());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Un utilisateur IAM ou un autre Compte AWS peut demander des informations d'identification de sécurité temporaires à l'aide du AWS SDK for .NET et les utiliser pour accéder à Amazon S3. Ces informations d'identification expirent à la fin de la session.

Par défaut, la session dure une heure. Si vous utilisez les informations d'identification des utilisateurs IAM, vous pouvez spécifier la durée (de 15 minutes à la durée maximale de session pour le rôle) lors de la demande d'informations d'identification de sécurité temporaires. Pour en savoir plus sur les informations d'identification de sécurité temporaires, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la création de demandes, consultez Demandes.

Pour obtenir des informations d'identification de sécurité temporaires et accéder à Amazon S3

1. Créez une instance du AWS Security Token Service client,`AmazonSecurityTokenServiceClient`. Pour en savoir plus sur les informations d'identification, consultez Développement avec Amazon S3 à l'aide des AWS SDK.
2. Lancez une session en appelant la méthode `GetSessionToken` du client STS que vous avez créé dans l'étape précédente. Vous fournissez des informations de session à cette méthode grâce à un objet `GetSessionTokenRequest`.

   La méthode renvoie les informations d'identification de sécurité temporaires.

3. Regroupez les informations d'identification de sécurité temporaires dans une instance de l'objet `SessionAWSCredentials`. Vous utilisez cet objet pour fournir les informations d'identification de sécurité temporaires à votre client Amazon S3.
4. Créez une instance de la classe `AmazonS3Client` en fournissant les informations d'identification de sécurité temporaires. Vous envoyez des demandes à Amazon S3 grâce à ce client. Si vous envoyez des demandes à l'aide d'informations d'identification expirées, Amazon S3 renvoie une erreur.

> **Note**
>
> Si vous obtenez les informations d'identification de sécurité temporaires à l'aide des informations d'identification de sécurité de votre Compte AWS , ces informations d'identification de sécurité temporaires ne sont valides que pendant une heure. Vous pouvez spécifier une durée de session uniquement si vous utilisez les informations d'identification des utilisateurs IAM pour demander une session.

L'exemple C# suivant répertorie les clés d'objet dans le compartiment spécifié. À titre d'illustration, l'exemple obtient des informations d'identification de sécurité temporaires pour une session d'une heure par défaut et les utilise pour envoyer une demande authentifiée à Amazon S3.

Si vous souhaitez tester l'exemple à l'aide des informations d'identification de l'utilisateur IAM, vous devez créer un utilisateur IAM sous votre Compte AWS. Pour de plus amples informations sur la création d'un utilisateur IAM, veuillez consulter Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la création de demandes, consultez Demandes.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
using System;
```

```
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TempCredExplicitSessionStartTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            ListObjectsAsync().Wait();
        }

        private static async Task ListObjectsAsync()
        {
            try
            {
                // Credentials use the default AWS SDK for .NET credential search
 chain.
                // On local development machines, this is your default profile.
                Console.WriteLine("Listing objects stored in a bucket");
                SessionAWSCredentials tempCredentials = await
 GetTemporaryCredentialsAsync();

                // Create a client by providing temporary security credentials.
                using (s3Client = new AmazonS3Client(tempCredentials, bucketRegion))
                {
                    var listObjectRequest = new ListObjectsRequest
                    {
                        BucketName = bucketName
                    };
                    // Send request to Amazon S3.
                    ListObjectsResponse response = await
 s3Client.ListObjectsAsync(listObjectRequest);
                    List<S3Object> objects = response.S3Objects;
                    Console.WriteLine("Object count = {0}", objects.Count);
                }
            }
            catch (AmazonS3Exception s3Exception)
            {
```

```
                Console.WriteLine(s3Exception.Message, s3Exception.InnerException);
            }
            catch (AmazonSecurityTokenServiceException stsException)
            {
                Console.WriteLine(stsException.Message,
stsException.InnerException);
            }
        }

        private static async Task<SessionAWSCredentials>
GetTemporaryCredentialsAsync()
        {
            using (var stsClient = new AmazonSecurityTokenServiceClient())
            {
                var getSessionTokenRequest = new GetSessionTokenRequest
                {
                    DurationSeconds = 7200 // seconds
                };

                GetSessionTokenResponse sessionTokenResponse =
                            await
stsClient.GetSessionTokenAsync(getSessionTokenRequest);

                Credentials credentials = sessionTokenResponse.Credentials;

                var sessionCredentials =
                    new SessionAWSCredentials(credentials.AccessKeyId,
                                              credentials.SecretAccessKey,
                                              credentials.SessionToken);
                return sessionCredentials;
            }
        }
    }
}
```

PHP

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Un utilisateur IAM ou un Compte AWS peut demander des informations d'identification de sécurité temporaires à l'aide de la version 3 du AWS SDK for PHP. Puis, il utilise les informations d'identification temporaires pour accéder à Amazon S3. Les informations d'identification expirent à la fin de la session.

Par défaut, la session dure une heure. Si vous utilisez les informations d'identification des utilisateurs IAM, vous pouvez spécifier la durée (de 15 minutes à la durée maximale de session pour le rôle) lors de la demande d'informations d'identification de sécurité temporaires. Pour en savoir plus sur les informations d'identification de sécurité temporaires, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la création de demandes, consultez Demandes.

> **Note**
>
> Si vous obtenez des informations d'identification de sécurité temporaires grâce aux informations d'identification de sécurité du Compte AWS , les informations d'identification de sécurité temporaires ne sont valides que pendant une heure. Vous pouvez spécifier la durée de session uniquement si vous utilisez les informations d'identification des utilisateurs IAM pour demander une session.

Example

L'exemple de PHP suivant liste les clés d'objet dans le compartiment spécifié grâce aux informations d'identification de sécurité temporaires. L'exemple obtient des informations d'identification de sécurité temporaires pour une session d'une heure par défaut, et les utilise pour envoyer une demande authentifiée à Amazon S3. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Si vous souhaitez tester l'exemple à l'aide des informations d'identification de l'utilisateur IAM, vous devez créer un utilisateur IAM sous votre. Compte AWS Pour plus d'informations sur la création d'un utilisateur IAM, consultez la section Création de votre premier utilisateur IAM et de votre premier groupe d'administrateurs dans le guide de l'utilisateur IAM. Pour obtenir des exemples de configuration de la durée de la session lors de l'utilisation des informations d'identification des utilisateurs IAM pour demander une session, veuillez consulter Envoi de demandes à l'aide des informations d'identification temporaires des utilisateurs IAM .

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;

$bucket = '*** Your Bucket Name ***';
```

```
$sts = new StsClient([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

$sessionToken = $sts->getSessionToken();

$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
    'credentials' => [
        'key'    => $sessionToken['Credentials']['AccessKeyId'],
        'secret' => $sessionToken['Credentials']['SecretAccessKey'],
        'token'  => $sessionToken['Credentials']['SessionToken']
    ]
]);

$result = $s3->listBuckets();

try {
    // Retrieve a paginator for listing objects.
    $objects = $s3->getPaginator('ListObjects', [
        'Bucket' => $bucket
    ]);

    echo "Keys retrieved!" . PHP_EOL;

    // List objects
    foreach ($objects as $object) {
        echo $object['Key'] . PHP_EOL;
    }
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Un utilisateur IAM ou un Compte AWS peut demander des informations d'identification de sécurité temporaires en les utilisant AWS SDK for Ruby et les utiliser pour accéder à Amazon S3. Ces informations d'identification expirent à la fin de la session.

Par défaut, la session dure une heure. Si vous utilisez les informations d'identification des utilisateurs IAM, vous pouvez spécifier la durée (de 15 minutes à la durée maximale de session pour le rôle) lors de la demande d'informations d'identification de sécurité temporaires. Pour en savoir plus sur les informations d'identification de sécurité temporaires, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la création de demandes, consultez Demandes.

> **Note**
>
> Si vous obtenez des informations d'identification de sécurité temporaires grâce aux informations d'identification de sécurité de votre Compte AWS , les informations d'identification de sécurité temporaires ne sont valides que pendant une heure. Vous pouvez spécifier une durée de session uniquement si vous utilisez les informations d'identification des utilisateurs IAM pour demander une session.

L'exemple de code Ruby suivant crée un utilisateur temporaire afin de répertorier les éléments d'un compartiment spécifié pendant une heure. Pour utiliser cet exemple, vous devez disposer AWS d'informations d'identification disposant des autorisations nécessaires pour créer de nouveaux AWS Security Token Service (AWS STS) clients et répertorier les buckets Amazon S3.

```
# Prerequisites:
# - A user in AWS Identity and Access Management (IAM). This user must
#   be able to assume the following IAM role. You must run this code example
#   within the context of this user.
# - An existing role in IAM that allows all of the Amazon S3 actions for all of the
#   resources in this code example. This role must also trust the preceding IAM
 user.
# - An existing S3 bucket.

require "aws-sdk-core"
require "aws-sdk-s3"
require "aws-sdk-iam"

# Checks whether a user exists in IAM.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [Boolean] true if the user exists; otherwise, false.
```

```
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   exit 1 unless user_exists?(iam_client, 'my-user')
def user_exists?(iam_client, user_name)
  response = iam_client.get_user(user_name: user_name)
  return true if response.user.user_name
rescue Aws::IAM::Errors::NoSuchEntity
  # User doesn't exist.
rescue StandardError => e
  puts "Error while determining whether the user " \
    "'#{user_name}' exists: #{e.message}"
end

# Creates a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS:IAM::Types::User] The new user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = create_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def create_user(iam_client, user_name)
  response = iam_client.create_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while creating the user '#{user_name}': #{e.message}"
end

# Gets a user in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [AWS:IAM::Types::User] The existing user.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   user = get_user(iam_client, 'my-user')
#   exit 1 unless user.user_name
def get_user(iam_client, user_name)
  response = iam_client.get_user(user_name: user_name)
  return response.user
rescue StandardError => e
  puts "Error while getting the user '#{user_name}': #{e.message}"
end
```

```
# Checks whether a role exists in IAM.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The role's name.
# @return [Boolean] true if the role exists; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-west-2')
#   exit 1 unless role_exists?(iam_client, 'my-role')
def role_exists?(iam_client, role_name)
  response = iam_client.get_role(role_name: role_name)
  return true if response.role.role_name
rescue StandardError => e
  puts "Error while determining whether the role " \
    "'#{role_name}' exists: #{e.message}"
end

# Gets credentials for a role in IAM.
#
# @param sts_client [Aws::STS::Client] An initialized AWS STS client.
# @param role_arn [String] The role's Amazon Resource Name (ARN).
# @param role_session_name [String] A name for this role's session.
# @param duration_seconds [Integer] The number of seconds this session is valid.
# @return [AWS::AssumeRoleCredentials] The credentials.
# @example
#   sts_client = Aws::STS::Client.new(region: 'us-west-2')
#   credentials = get_credentials(
#     sts_client,
#     'arn:aws:iam::123456789012:role/AmazonS3ReadOnly',
#     'ReadAmazonS3Bucket',
#     3600
#   )
#   exit 1 if credentials.nil?
def get_credentials(sts_client, role_arn, role_session_name, duration_seconds)
  Aws::AssumeRoleCredentials.new(
    client: sts_client,
    role_arn: role_arn,
    role_session_name: role_session_name,
    duration_seconds: duration_seconds
  )
rescue StandardError => e
  puts "Error while getting credentials: #{e.message}"
end
```

```
# Checks whether a bucket exists in Amazon S3.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The name of the bucket.
# @return [Boolean] true if the bucket exists; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless bucket_exists?(s3_client, 'doc-example-bucket')
def bucket_exists?(s3_client, bucket_name)
  response = s3_client.list_buckets
  response.buckets.each do |bucket|
    return true if bucket.name == bucket_name
  end
rescue StandardError => e
  puts "Error while checking whether the bucket '#{bucket_name}' " \
    "exists: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'doc-example-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if response.count.positive?
    puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
    puts "Name => ETag"
    response.contents.each do |obj|
      puts "#{obj.key} => #{obj.etag}"
    end
  else
    puts "No objects in the bucket named '#{bucket_name}'."
  end
  return true
rescue StandardError => e
```

```
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end
```

Ressources connexes

- Développement avec Amazon S3 à l'aide des AWS SDK
- AWS SDK for PHP pour la classe Amazon S3 Aws \ S3 \ S3Client
- Documentation AWS SDK for PHP

Demandes grâce aux informations d'identification de sécurité temporaires de l'utilisateur fédéré

Vous pouvez demander des informations d'identification de sécurité temporaires et les fournir à vos utilisateurs fédérés ou aux applications qui ont besoin d'accéder à vos AWS ressources. Cette section fournit des exemples de la manière dont vous pouvez utiliser le AWS SDK pour obtenir des informations d'identification de sécurité temporaires pour vos utilisateurs fédérés ou vos applications et envoyer des demandes authentifiées à Amazon S3 à l'aide de ces informations d'identification. Pour obtenir la liste des AWS SDK disponibles, consultez la section Exemples de code et de bibliothèques.

> **Note**
>
> L'utilisateur Compte AWS et un utilisateur IAM peuvent demander des informations d'identification de sécurité temporaires pour les utilisateurs fédérés. Toutefois, par mesure de sécurité, seul un utilisateur IAM muni des autorisations nécessaires peut demander ces informations d'identification de sécurité temporaires, pour veiller à ce que l'utilisateur fédéré obtienne autant d'autorisations que l'utilisateur IAM. Dans certaines applications, vous choisirez peut-être de créer un utilisateur IAM avec des autorisations spécifiques, afin d'accorder uniquement des informations d'identification de sécurité temporaires à vos utilisateurs fédérés et vos applications.

Java

Vous pouvez fournir des informations d'identification de sécurité temporaires à vos utilisateurs fédérés et à vos applications afin qu'ils puissent envoyer des demandes authentifiées pour accéder à vos AWS ressources. Lorsque vous demandez ces informations d'identification de sécurité temporaires, vous devez fournir un nom d'utilisateur et une stratégie IAM décrivant les autorisations de ressources que vous souhaitez accorder. Par défaut, la session dure une heure. Vous pouvez explicitement définir une valeur de durée différente lorsque vous demandez des informations d'identification de sécurité temporaires pour les utilisateurs fédérés et les applications.

> **Note**
>
> Lorsque vous demandez des informations d'identification de sécurité temporaires pour des utilisateurs fédérés et des applications, nous vous recommandons, pour plus de

sécurité, d'utiliser un utilisateur IAM dédié disposant uniquement des autorisations d'accès nécessaires. L'utilisateur temporaire que vous créez ne peut pas obtenir plus d'autorisations que l'utilisateur IAM ayant demandé les informations d'identification de sécurité temporaires. Pour de plus amples informations, veuillez consulter Questions fréquentes (FAQ)AWS Identity and Access Management.

Pour fournir des informations d'identification de sécurité et envoyer une demande authentifiée pour accéder aux ressources, procédez comme suit :

- Créez une instance de la classe `AWSSecurityTokenServiceClient`.
- Lancez une session en appelant la méthode `getFederationToken()` du client STS (Security Token Service). Fournissez des informations de session, y compris le nom d'utilisateur et une stratégie IAM, que vous souhaitez attacher aux informations d'identification temporaires. Vous pouvez renseigner une durée de session optionnelle. Cette méthode renvoie les informations d'identification de sécurité temporaires.
- Regroupez les informations d'identification de sécurité temporaires dans une instance de l'objet `BasicSessionCredentials`. Vous utilisez cet objet pour fournir les informations d'identification de sécurité temporaires à votre client Amazon S3.
- Créez une instance de la classe `AmazonS3Client` à l'aide des informations d'identification de sécurité temporaires. Vous envoyez des demandes à Amazon S3 grâce à ce client. Si vous envoyez des demandes à l'aide d'informations d'identification expirées, Amazon S3 renvoie une erreur.

Example

L'exemple répertorie les clés dans le compartiment S3 spécifié. Dans l'exemple, vous obtenez des informations d'identification de sécurité temporaires pour une session de deux heures pour votre utilisateur fédéré et les utilisez pour envoyer des demandes authentifiées à Amazon S3. Pour exécuter cet exemple, vous devez créer un utilisateur IAM avec une politique attachée qui permet à l'utilisateur de demander des informations d'identification de sécurité temporaires et de répertorier vos AWS ressources. La stratégie suivante permet d'effectuer ceci :

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
```

```
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

Pour de plus amples informations sur la création d'un utilisateur IAM, veuillez consulter Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

Après avoir créé un utilisateur IAM et attaché la stratégie précédente, vous pouvez exécuter l'exemple suivant. Pour obtenir des instructions sur la création et le test d'un échantillon de travail, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.auth.policy.Policy;
import com.amazonaws.auth.policy.Resource;
import com.amazonaws.auth.policy.Statement;
import com.amazonaws.auth.policy.Statement.Effect;
import com.amazonaws.auth.policy.actions.S3Actions;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectListing;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.Credentials;
import com.amazonaws.services.securitytoken.model.GetFederationTokenRequest;
import com.amazonaws.services.securitytoken.model.GetFederationTokenResult;

import java.io.IOException;

public class MakingRequestsWithFederatedTempCredentials {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Specify bucket name ***";
        String federatedUser = "*** Federated user name ***";
```

```
        String resourceARN = "arn:aws:s3:::" + bucketName;

        try {
            AWSSecurityTokenService stsClient = AWSSecurityTokenServiceClientBuilder
                    .standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            GetFederationTokenRequest getFederationTokenRequest = new
GetFederationTokenRequest();
            getFederationTokenRequest.setDurationSeconds(7200);
            getFederationTokenRequest.setName(federatedUser);

            // Define the policy and add it to the request.
            Policy policy = new Policy();
            policy.withStatements(new Statement(Effect.Allow)
                    .withActions(S3Actions.ListObjects)
                    .withResources(new Resource(resourceARN)));
            getFederationTokenRequest.setPolicy(policy.toJson());

            // Get the temporary security credentials.
            GetFederationTokenResult federationTokenResult =
stsClient.getFederationToken(getFederationTokenRequest);
            Credentials sessionCredentials = federationTokenResult.getCredentials();

            // Package the session credentials as a BasicSessionCredentials
            // object for an Amazon S3 client object to use.
            BasicSessionCredentials basicSessionCredentials = new
BasicSessionCredentials(
                    sessionCredentials.getAccessKeyId(),
                    sessionCredentials.getSecretAccessKey(),
                    sessionCredentials.getSessionToken());
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new
AWSStaticCredentialsProvider(basicSessionCredentials))
                    .withRegion(clientRegion)
                    .build();

            // To verify that the client works, send a listObjects request using
            // the temporary security credentials.
            ObjectListing objects = s3Client.listObjects(bucketName);
            System.out.println("No. of Objects = " +
objects.getObjectSummaries().size());
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Vous pouvez fournir des informations d'identification de sécurité temporaires à vos utilisateurs fédérés et à vos applications afin qu'ils puissent envoyer des demandes authentifiées pour accéder à vos AWS ressources. Lorsque vous demandez ces informations d'identification de sécurité temporaires, vous devez fournir un nom d'utilisateur et une stratégie IAM décrivant les autorisations de ressources que vous souhaitez accorder. Par défaut, la session dure une heure. Vous pouvez explicitement définir une valeur de durée différente lorsque vous demandez des informations d'identification de sécurité temporaires pour les utilisateurs fédérés et les applications. Pour obtenir des informations sur l'envoi de demandes authentifiées, consultez Demandes.

> **Note**
>
> Lorsque vous demandez des informations d'identification de sécurité temporaires pour des utilisateurs fédérés et des applications, nous vous suggérons, pour plus de sécurité, d'utiliser un utilisateur IAM dédié disposant uniquement des autorisations d'accès nécessaires. L'utilisateur temporaire que vous créez ne peut pas obtenir plus d'autorisations que l'utilisateur IAM ayant demandé les informations d'identification de sécurité temporaires. Pour de plus amples informations, veuillez consulter Questions fréquentes (FAQ)AWS Identity and Access Management.

Vous effectuez les actions suivantes :

- Créez une instance du AWS Security Token Service client, `AmazonSecurityTokenServiceClient` classe.

- Lancez une session en invoquant la méthode `GetFederationToken` du client STS. Vous devez fournir des informations de session, y compris le nom d'utilisateur et une stratégie IAM, que vous souhaitez attacher aux informations d'identification temporaires. Vous pouvez également renseigner une durée de session. Cette méthode renvoie les informations d'identification de sécurité temporaires.
- Regroupez les informations d'identification de sécurité temporaires dans une instance de l'objet `SessionAWSCredentials`. Vous utilisez cet objet pour fournir les informations d'identification de sécurité temporaires à votre client Amazon S3.
- Créez une instance de la classe `AmazonS3Client` en fournissant les informations d'identification de sécurité temporaires. Vous utilisez ce client pour envoyer des demandes à Amazon S3. Si vous envoyez des demandes à l'aide d'informations d'identification expirées, Amazon S3 renvoie une erreur.

Example

L'exemple C# suivant répertorie les clés dans le compartiment spécifié. Dans l'exemple, vous obtenez des informations d'identification de sécurité temporaires pour une session de deux heures pour votre utilisateur fédéré (Utilisateur 1), et les utilisez pour envoyer des demandes authentifiées à Amazon S3.

- Pour cet exercice, vous créez un utilisateur IAM disposant d'autorisations minimales. À l'aide des informations d'identification de cet utilisateur IAM, vous demandez des informations d'identification temporaires pour d'autres utilisateurs. Cet exemple répertorie uniquement les objets contenus dans un compartiment spécifique. Créez un utilisateur IAM auquel est attachée la stratégie suivante :

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

La politique permet à l'utilisateur IAM de demander des informations d'identification de sécurité temporaires et une autorisation d'accès uniquement pour répertorier vos AWS ressources. Pour de plus amples informations sur la création d'un utilisateur IAM, veuillez consulter Création de votre groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

- Utilisez les informations d'identification de sécurité des utilisateurs IAM pour tester l'exemple suivant. Cet exemple envoie une demande authentifiée à Amazon S3 à l'aide d'informations d'identification de sécurité temporaires. L'exemple indique la stratégie suivante lors de la demande des informations d'identification de sécurité temporaires pour l'utilisateur fédéré (User1), qui limite l'accès aux objets répertoriés dans un compartiment spécifique (`YourBucketName`). Vous devez mettre à jour la stratégie et fournir votre propre nom de compartiment existant.

```
{
  "Statement":[
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
    }
  ]
}
```

- Example

  Mettez à jour l'exemple suivant et fournissez le nom de compartiment que vous avez spécifié dans la stratégie d'accès d'utilisateur fédéré précédente. Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.SecurityToken;
using Amazon.SecurityToken.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class TempFederatedCredentialsTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            ListObjectsAsync().Wait();
        }

        private static async Task ListObjectsAsync()
        {
            try
            {
                Console.WriteLine("Listing objects stored in a bucket");
                // Credentials use the default AWS SDK for .NET credential search
 chain.
                // On local development machines, this is your default profile.
                SessionAWSCredentials tempCredentials =
                    await GetTemporaryFederatedCredentialsAsync();

                // Create a client by providing temporary security credentials.
                using (client = new AmazonS3Client(bucketRegion))
                {
                    ListObjectsRequest listObjectRequest = new
 ListObjectsRequest();
                    listObjectRequest.BucketName = bucketName;

                    ListObjectsResponse response = await
 client.ListObjectsAsync(listObjectRequest);
                    List<S3Object> objects = response.S3Objects;
                    Console.WriteLine("Object count = {0}", objects.Count);

                    Console.WriteLine("Press any key to continue...");
                    Console.ReadKey();
                }
            }
            catch (AmazonS3Exception e)
            {
```

```
                Console.WriteLine("Error encountered ***. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}'
when writing an object", e.Message);
            }
        }

        private static async Task<SessionAWSCredentials>
GetTemporaryFederatedCredentialsAsync()
        {
            AmazonSecurityTokenServiceConfig config = new
AmazonSecurityTokenServiceConfig();
            AmazonSecurityTokenServiceClient stsClient =
                new AmazonSecurityTokenServiceClient(
                                            config);

            GetFederationTokenRequest federationTokenRequest =
                                      new GetFederationTokenRequest();
            federationTokenRequest.DurationSeconds = 7200;
            federationTokenRequest.Name = "User1";
            federationTokenRequest.Policy = @"{
               ""Statement"":
               [
                 {
                   ""Sid"":""Stmt1311212314284"",
                   ""Action"":[""s3:ListBucket""],
                   ""Effect"":""Allow"",
                   ""Resource"":""arn:aws:s3:::" + bucketName + @"""
                 }
               ]
             }
            ";

            GetFederationTokenResponse federationTokenResponse =
                        await
stsClient.GetFederationTokenAsync(federationTokenRequest);
            Credentials credentials = federationTokenResponse.Credentials;

            SessionAWSCredentials sessionCredentials =
                new SessionAWSCredentials(credentials.AccessKeyId,
                                          credentials.SecretAccessKey,
```

```
                                            credentials.SessionToken);
            return sessionCredentials;
        }
    }
}
```

PHP

Cette rubrique explique comment utiliser les classes de la version 3 de AWS SDK for PHP pour demander des informations d'identification de sécurité temporaires pour les utilisateurs fédérés et les applications et les utiliser pour accéder aux ressources stockées dans Amazon S3. Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Vous pouvez fournir des informations d'identification de sécurité temporaires à vos utilisateurs fédérés et à vos applications afin qu'ils puissent envoyer des demandes authentifiées pour accéder à vos AWS ressources. Lorsque vous demandez ces informations d'identification de sécurité temporaires, vous devez fournir un nom d'utilisateur et une stratégie IAM décrivant les autorisations de ressources que vous souhaitez accorder. Ces informations d'identification expirent à la fin de la session. Par défaut, la session dure une heure. Vous pouvez explicitement définir une valeur de durée différente lorsque vous demandez des informations d'identification de sécurité temporaires pour les utilisateurs fédérés et les applications. Pour en savoir plus sur les informations d'identification de sécurité temporaires, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la transmission d'informations d'identification de sécurité temporaires à vos utilisateurs fédérés et aux applications, consultez Demandes.

Lorsque vous demandez des informations d'identification de sécurité temporaires pour des utilisateurs fédérés et des applications, nous vous recommandons, pour plus de sécurité, d'utiliser un utilisateur IAM dédié disposant uniquement des autorisations d'accès nécessaires. L'utilisateur temporaire que vous créez ne peut pas obtenir plus d'autorisations que l'utilisateur IAM ayant demandé les informations d'identification de sécurité temporaires. Pour de plus amples informations sur la fédération d'identité, veuillez consulter FAQ AWS Identity and Access Management.

Pour plus d'informations sur l'API AWS SDK for Ruby, consultez AWS SDK for Ruby - Version 2.

Example

L'exemple de PHP suivant répertorie les clés dans le compartiment spécifié. Dans l'exemple, vous obtenez des informations d'identification de sécurité temporaires pour une session d'une heure pour votre utilisateur fédéré (User1). Vous utilisez ensuite les informations d'identification de sécurité temporaires pour envoyer les demandes authentifiées à Amazon S3.

Pour plus de sécurité, lorsque vous demandez des informations d'identification de sécurité temporaires pour d'autres utilisateurs, vous utilisez les informations d'identification d'un utilisateur IAM disposant des autorisations pour demander des informations d'identification de sécurité temporaires. Pour veiller à ce que cet utilisateur IAM accorde uniquement à l'utilisateur fédéré les autorisations minimales propres aux applications, vous pouvez également limiter les autorisations d'accès de cet utilisateur IAM. Cet exemple répertorie uniquement les objets contenus dans un compartiment spécifique. Créez un utilisateur IAM auquel est attachée la stratégie suivante :

```
{
  "Statement":[{
      "Action":["s3:ListBucket",
        "sts:GetFederationToken*"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

La politique permet à l'utilisateur IAM de demander des informations d'identification de sécurité temporaires et une autorisation d'accès uniquement pour répertorier vos AWS ressources. Pour de plus amples informations sur la création d'un utilisateur IAM, veuillez consulter Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

Vous pouvez désormais utiliser les informations d'identification de sécurité des utilisateurs IAM pour tester l'exemple suivant. Cet exemple envoie une demande authentifiée à Amazon S3 à l'aide d'informations d'identification de sécurité temporaires. Lors d'une demande d'informations d'identification de sécurité temporaires pour l'utilisateur fédéré (User1), cet exemple spécifie la stratégie suivante, qui limite l'accès pour répertorier les objets contenus dans un compartiment spécifique. Mettez à jour la stratégie avec le nom de votre compartiment.

```
{
  "Statement":[
```

```
    {
      "Sid":"1",
      "Action":["s3:ListBucket"],
      "Effect":"Allow",
      "Resource":"arn:aws:s3:::YourBucketName"
    }
  ]
}
```

Dans l'exemple suivant, remplacez `YourBucketName` par le nom de votre compartiment lorsque vous spécifiez une ressource de stratégie.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;
use Aws\Sts\StsClient;

$bucket = '*** Your Bucket Name ***';

// In real applications, the following code is part of your trusted code. It has
// the security credentials that you use to obtain temporary security credentials.
$sts = new StsClient([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Fetch the federated credentials.
$sessionToken = $sts->getFederationToken([
    'Name'              => 'User1',
    'DurationSeconds'    => '3600',
    'Policy'            => json_encode([
        'Statement' => [
            'Sid'               => 'randomstatementid' . time(),
            'Action'            => ['s3:ListBucket'],
            'Effect'            => 'Allow',
            'Resource'          => 'arn:aws:s3:::' . $bucket
        ]
    ])
]);

// The following will be part of your less trusted code. You provide temporary
// security credentials so the code can send authenticated requests to Amazon S3.
```

```
$s3 = new S3Client([
    'region' => 'us-east-1',
    'version' => 'latest',
    'credentials' => [
        'key' => $sessionToken['Credentials']['AccessKeyId'],
        'secret' => $sessionToken['Credentials']['SecretAccessKey'],
        'token' => $sessionToken['Credentials']['SessionToken']
    ]
]);

try {
    $result = $s3->listObjects([
        'Bucket' => $bucket
    ]);
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Vous pouvez fournir des informations d'identification de sécurité temporaires à vos utilisateurs fédérés et à vos applications afin qu'ils puissent envoyer des demandes authentifiées pour accéder à vos AWS ressources. Lorsque vous demandez des informations d'identification de sécurité temporaires auprès du service IAM, vous devez fournir un nom d'utilisateur et une stratégie IAM décrivant les autorisations de ressources que vous souhaitez accorder. Par défaut, la session dure une heure. Toutefois, si vous demandez des informations d'identification temporaires à l'aide des informations d'identification des utilisateurs IAM, vous pouvez explicitement définir une valeur de durée différente lorsque vous demandez les informations d'identification de sécurité temporaires pour des utilisateurs fédérés et des applications. Pour plus d'informations sur les informations d'identification de sécurité temporaires pour vos utilisateurs fédérés et applications, consultez Demandes.

> **Note**
>
> Lorsque vous demandez des informations d'identification de sécurité temporaires pour des utilisateurs fédérés et des applications, vous souhaiterez peut-être utiliser un utilisateur IAM dédié disposant uniquement des autorisations d'accès nécessaires. L'utilisateur temporaire que vous créez ne peut pas obtenir plus d'autorisations que l'utilisateur IAM ayant demandé les informations d'identification de sécurité temporaires.

Pour de plus amples informations, veuillez consulter Questions fréquentes (FAQ)AWS Identity and Access Management.

Example

L'exemple de code Ruby suivant autorise un utilisateur fédéré disposant d'un ensemble limité d'autorisations à répertorier des clés dans le compartiment spécifié.

```
# Prerequisites:
#  - An existing Amazon S3 bucket.

require "aws-sdk-s3"
require "aws-sdk-iam"
require "json"

# Checks to see whether a user exists in IAM; otherwise,
# creates the user.
#
# @param iam [Aws::IAM::Client] An initialized IAM client.
# @param user_name [String] The user's name.
# @return [Aws::IAM::Types::User] The existing or new user.
# @example
#   iam = Aws::IAM::Client.new(region: 'us-west-2')
#   user = get_user(iam, 'my-user')
#   exit 1 unless user.user_name
#   puts "User's name: #{user.user_name}"
def get_user(iam, user_name)
  puts "Checking for a user with the name '#{user_name}'..."
  response = iam.get_user(user_name: user_name)
  puts "A user with the name '#{user_name}' already exists."
  return response.user
# If the user doesn't exist, create them.
rescue Aws::IAM::Errors::NoSuchEntity
  puts "A user with the name '#{user_name}' doesn't exist. Creating this user..."
  response = iam.create_user(user_name: user_name)
  iam.wait_until(:user_exists, user_name: user_name)
  puts "Created user with the name '#{user_name}'."
  return response.user
rescue StandardError => e
  puts "Error while accessing or creating the user named '#{user_name}':
 #{e.message}"
end
```

```
# Gets temporary AWS credentials for an IAM user with the specified permissions.
#
# @param sts [Aws::STS::Client] An initialized AWS STS client.
# @param duration_seconds [Integer] The number of seconds for valid credentials.
# @param user_name [String] The user's name.
# @param policy [Hash] The access policy.
# @return [Aws::STS::Types::Credentials] AWS credentials for API authentication.
# @example
#   sts = Aws::STS::Client.new(region: 'us-west-2')
#   credentials = get_temporary_credentials(sts, duration_seconds, user_name,
#     {
#       'Version' => '2012-10-17',
#       'Statement' => [
#         'Sid' => 'Stmt1',
#         'Effect' => 'Allow',
#         'Action' => 's3:ListBucket',
#         'Resource' => 'arn:aws:s3:::doc-example-bucket'
#       ]
#     }
#   )
#   exit 1 unless credentials.access_key_id
#   puts "Access key ID: #{credentials.access_key_id}"
def get_temporary_credentials(sts, duration_seconds, user_name, policy)
  response = sts.get_federation_token(
    duration_seconds: duration_seconds,
    name: user_name,
    policy: policy.to_json
  )
  return response.credentials
rescue StandardError => e
  puts "Error while getting federation token: #{e.message}"
end

# Lists the keys and ETags for the objects in an Amazon S3 bucket.
#
# @param s3_client [Aws::S3::Client] An initialized Amazon S3 client.
# @param bucket_name [String] The bucket's name.
# @return [Boolean] true if the objects were listed; otherwise, false.
# @example
#   s3_client = Aws::S3::Client.new(region: 'us-west-2')
#   exit 1 unless list_objects_in_bucket?(s3_client, 'doc-example-bucket')
def list_objects_in_bucket?(s3_client, bucket_name)
  puts "Accessing the contents of the bucket named '#{bucket_name}'..."
```

```
  response = s3_client.list_objects_v2(
    bucket: bucket_name,
    max_keys: 50
  )

  if response.count.positive?
    puts "Contents of the bucket named '#{bucket_name}' (first 50 objects):"
    puts "Name => ETag"
    response.contents.each do |obj|
      puts "#{obj.key} => #{obj.etag}"
    end
  else
    puts "No objects in the bucket named '#{bucket_name}'."
  end
  return true
rescue StandardError => e
  puts "Error while accessing the bucket named '#{bucket_name}': #{e.message}"
end

# Example usage:
def run_me
  region = "us-west-2"
  user_name = "my-user"
  bucket_name = "doc-example-bucket"

  iam = Aws::IAM::Client.new(region: region)
  user = get_user(iam, user_name)

  exit 1 unless user.user_name

  puts "User's name: #{user.user_name}"
  sts = Aws::STS::Client.new(region: region)
  credentials = get_temporary_credentials(sts, 3600, user_name,
    {
      "Version" => "2012-10-17",
      "Statement" => [
        "Sid" => "Stmt1",
        "Effect" => "Allow",
        "Action" => "s3:ListBucket",
        "Resource" => "arn:aws:s3:::#{bucket_name}"
      ]
    }
  )
```

```
  exit 1 unless credentials.access_key_id

  puts "Access key ID: #{credentials.access_key_id}"
  s3_client = Aws::S3::Client.new(region: region, credentials: credentials)

  exit 1 unless list_objects_in_bucket?(s3_client, bucket_name)
end

run_me if $PROGRAM_NAME == __FILE__
```

Ressources connexes

- Développement avec Amazon S3 à l'aide des AWS SDK
- AWS SDK for PHP pour la classe Amazon S3 Aws \ S3 \ S3Client
- Documentation AWS SDK for PHP

Demandes à l'aide de l'API REST

Cette section contient des informations expliquant comment envoyer des demandes à des points de terminaison Amazon S3 à l'aide de l'API REST. Pour obtenir la liste des points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.

Construction de noms d'hôte S3 pour les demandes d'API REST

Les points de terminaison Amazon S3 suivent la structure indiquée ci-dessous :

```
s3.Region.amazonaws.com
```

Les points de terminaison de points d'accès Amazon S3 et les points de terminaison Dual-Stack (double pile) suivent également la structure standard :

- Points d'accès Amazon S -`s3-accesspoint.Region.amazonaws.com`
- Dual-Stack (double pile - `s3.dualstack.Region.amazonaws.com`

Pour obtenir la liste complète des régions et points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans le Référence générale d'Amazon Web Services.

Demandes de type hébergement virtuel et de type chemin

Lorsque vous envoyez des demandes à l'aide de l'API REST, vous pouvez utiliser des URI de types hébergement virtuel ou chemin pour les points de terminaison Amazon S3. Pour de plus amples informations, veuillez consulter Hébergement virtuel de compartiments.

Example Demande de type hébergement virtuel

Voici un exemple de demande de type hébergement virtuel visant à supprimer le fichier `puppy.jpg` du compartiment `examplebucket` dans la Région USA Ouest (Oregon). Pour en savoir plus sur les demandes d'hébergement virtuel, consultez Demandes de type hébergement virtuel.

```
DELETE /puppy.jpg HTTP/1.1
Host: examplebucket.s3.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Demande de type chemin

Voici un exemple d'une version de type chemin de la même demande.

```
DELETE /examplebucket/puppy.jpg HTTP/1.1
Host: s3.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Amazon S3 prend actuellement en charge l'accès aux URL de type hébergement virtuel et chemin d'accès dans toutes les Régions AWS. Toutefois, les URL de type chemin d'accès seront abandonnées à l'avenir. Pour plus d'informations, consultez la remarque importante suivante.

Pour de plus amples informations sur les demandes de type chemin, veuillez consulter Demandes de type chemin d'accès.

> **Important**
>
> Mise à jour (23 septembre 2020) – Pour laisser aux clients le temps nécessaire pour passer à des URL de type hébergement virtuel, nous avons décidé de retarder l'obsolescence des URL de type chemin d'accès. Pour de plus amples informations, veuillez consulter

Amazon S3 Path Deprecation Plan – The Rest of the Story dans le blog dédié aux actualités d'AWS.

Envoi de demandes à des points de terminaison Dual-Stack à l'aide de l'API REST

Lorsque vous utilisez l'API REST, vous pouvez accéder directement à un point de terminaison Dual-Stack (double pile) si vous utilisez un nom de point de terminaison (URI) de type hébergement virtuel ou chemin. Tous les noms de points de terminaison Dual-Stack (double pile) Amazon S3 incluent la Région concernée. Contrairement aux points de terminaison IPv4 standard, les points de terminaison de type hébergement virtuel ou chemin ont tous des noms spécifiques aux Régions.

Example Demande de point de terminaison Dual-Stack (double pile) de type hébergement virtuel

Vous pouvez utiliser un point de terminaison de type hébergement virtuel dans votre demande REST, comme indiqué dans l'exemple suivant, qui permet de récupérer l'objet `puppy.jpg` à partir du compartiment `examplebucket` dans la Région USA Ouest (Oregon).

```
GET /puppy.jpg HTTP/1.1
Host: examplebucket.s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Example Demande de point de terminaison Dual-Stack de type chemin

Vous pouvez également utiliser un point de terminaison de type chemin dans votre demande, comme indiqué dans l'exemple ci-dessous.

```
GET /examplebucket/puppy.jpg HTTP/1.1
Host: s3.dualstack.us-west-2.amazonaws.com
Date: Mon, 11 Apr 2016 12:00:00 GMT
x-amz-date: Mon, 11 Apr 2016 12:00:00 GMT
Authorization: authorization string
```

Pour de plus amples informations sur les points de terminaison Dual-Stack, consultez la section Utilisation des points de terminaison Dual-Stack Amazon S3.

Pour plus d'informations sur la procédure permettant d'envoyer des demandes à l'aide de l'API REST, consultez les rubriques ci-dessous.

Rubriques

Hébergement virtuel de compartiments

L'hébergement virtuel consiste à servir plusieurs sites web à partir d'un seul serveur web. Une façon de différencier les sites dans vos demandes d'API REST Amazon S3 consiste à utiliser le nom d'hôte apparent du Request-URI plutôt que simplement la partie du nom du chemin de l'URI. Une demande REST Amazon S3 ordinaire spécifie un compartiment en utilisant le premier composant délimité par une barre oblique du chemin de l'URI de demande. Sinon, vous pouvez utiliser l'hébergement virtuel Amazon S3 pour accéder à un compartiment dans un appel d'API REST grâce à l'en-tête HTTP `Host`. En pratique, Amazon S3 interprète `Host` en ce sens que la plupart des compartiments sont automatiquement accessibles (pour un type limité de demandes) à l'adresse `https://`*`bucket-name`*`.s3.`*`region-code`*`.amazonaws.com`. Pour obtenir la liste complète des régions et points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans le Référence générale d'Amazon Web Services.

L'hébergement virtuel présente d'autres avantages. Par ailleurs, en nommant le compartiment après avoir enregistré le nom de domaine et en faisant de ce nom un alias DNS pour Amazon S3, vous pouvez totalement personnaliser l'URL des ressources Amazon S3, par exemple, `http://my.`*`bucket-name`*`.com/`. Vous pouvez également publier dans le « répertoire racine » du serveur virtuel de votre compartiment. Cet avantage peut être considérable car beaucoup d'applications existantes recherchent des fichiers dans cet emplacement standard. Par exemple, on peut s'attendre à ce que les fichiers `favicon.ico`, `robots.txt` et `crossdomain.xml` se trouvent tous à la racine.

> **Important**
>
> Lorsque vous utilisez des compartiments de type hébergement virtuel avec SSL, le certificat générique SSL correspond uniquement aux compartiments qui ne contiennent pas de points (`.`). Pour contourner cette limitation, utilisez HTTP ou écrivez votre propre logique de vérification de certificat. Pour de plus amples informations, veuillez consulter Amazon S3 Path Deprecation Plan (Plan d'obsolescence de chemin d'accès Amazon S3) sur la page AWS News Blog.

Rubriques

Demandes de type chemin d'accès

Amazon S3 prend actuellement en charge l'accès aux URL de type hébergement virtuel et chemin d'accès dans toutes les Régions AWS. Toutefois, les URL de type chemin d'accès seront abandonnées à l'avenir. Pour plus d'informations, consultez la remarque importante suivante.

Dans Amazon S3, les URL de type chemin d'accès utilisent le format suivant :

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Par exemple, si vous créez un compartiment nommé `DOC-EXAMPLE-BUCKET1` dans la Région USA Ouest (Oregon) et que vous souhaitez accéder à l'objet `puppy.jpg` dans ce compartiment, vous pouvez utiliser l'URL de type chemin suivant :

```
https://s3.us-west-2.amazonaws.com/DOC-EXAMPLE-BUCKET1/puppy.jpg
```

> **Important**
>
> Mise à jour (23 septembre 2020) – Pour laisser aux clients le temps nécessaire pour passer à des URL de type hébergement virtuel, nous avons décidé de retarder l'obsolescence des URL de type chemin d'accès. Pour de plus amples informations, veuillez consulter Amazon S3 Path Deprecation Plan – The Rest of the Story dans le blog dédié aux actualités d'AWS.

> **Warning**
>
> Lorsque vous hébergez du contenu d'un site web auquel vous pourrez accéder à partir d'un navigateur web, évitez d'utiliser des URL de type chemin, car cela pourrait interférer avec le modèle de sécurité d'origine du navigateur. Pour héberger le contenu d'un site web, nous vous recommandons d'utiliser des points de terminaison de site web S3 ou une distribution CloudFront. Pour plus d'informations, consultez Points de terminaison de sites web et Deploy a React-based single-page application to Amazon S3 and CloudFront (Déployer une application d'une page basée sur React sur Amazon S3 et CloudFront) dans les Motifs AWS Perspective Guidance.

Demandes de type hébergement virtuel

Dans un URI de type hébergement virtuel, le nom du compartiment fait partie du nom de domaine compris dans l'URL.

Les URL de type hébergement virtuel Amazon S3 utilisent le format suivant :

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

Dans cet exemple, `DOC-EXAMPLE-BUCKET1` représente le nom de compartiment, USA Ouest (Oregon) représente la Région et `puppy.png` est le nom de la clé :

```
https://DOC-EXAMPLE-BUCKET1.s3.us-west-2.amazonaws.com/puppy.png
```

Spécification d'un compartiment d'en-tête **`Host`** HTTP

Tant que la demande `GET` n'utilise pas le point de terminaison SSL, vous pouvez spécifier le compartiment pour la demande grâce à l'en-tête `Host` HTTP. L'en-tête `Host` dans la demande REST est interprété comme suit :

- Si l'en-tête `Host` est omis ou que sa valeur est `s3.region-code.amazonaws.com`, le compartiment de la demande est le premier composant délimité par une barre oblique de l'URI de demande, et la clé de la demande constitue le reste de l'URI de demande. Il s'agit de la méthode ordinaire, comme illustré par le premier et le deuxième exemples de cette section. L'omission de l'en-tête `Host` est valide uniquement pour les demandes HTTP 1.0.
- Sinon, si la valeur de l'en-tête `Host` se termine par `.s3.region-code.amazonaws.com`, le nom du compartiment est le composant de tête de la valeur de l'en-tête `Host` jusqu'à `.s3.region-`

code.amazonaws.com. La clé de la demande est l'URI de demande. Cette interprétation expose les compartiments en tant que sous-domaines de .s3.*region-code*.amazonaws.com, comme illustré par les troisième et quatrième exemples de cette section.

- Sinon, le compartiment pour la demande est la valeur en minuscule de l'en-tête Host, et la clé pour la demande est l'URI de demande. Cette interprétation est utile lorsque vous avez enregistré le même nom DNS que votre nom de compartiment et que vous avez configuré ce nom comme alias de nom canonique (CNAME) pour Amazon S3. La procédure pour enregistrer des noms de domaine et configurer des enregistrements DNS CNAME n'entre pas dans le champ d'application de ce guide, mais le résultat est illustré par l'exemple final de cette section.

Exemples

Cette section fournit des exemples d'URL et de demandes.

Example – URL et demandes de type chemin d'accès

Cet exemple utilise les éléments suivants :

- Nom de compartiment - example.com
- Région - USA Est (Virginie du Nord)
- Nom de clé - homepage.html

L'URL se présente comme suit :

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

La demande se présente comme suit :

```
GET /example.com/homepage.html HTTP/1.1
Host: s3.us-east-1.amazonaws.com
```

La demande avec HTTP 1.0 et l'omission de l'en-tête Host se présentent comme suit :

```
GET /example.com/homepage.html HTTP/1.0
```

Pour en savoir plus sur les noms compatibles avec le DNS, consultez Limitations. Pour en savoir plus sur les clés, consultez Clés.

Example – Demandes et URL de type hébergement virtuel

Cet exemple utilise les éléments suivants :

- Nom de compartiment : `DOC-EXAMPLE-BUCKET1`
- Région : UE (Irlande)
- Nom de clé : `homepage.html`

L'URL se présente comme suit :

```
http://DOC-EXAMPLE-BUCKET1.s3.eu-west-1.amazonaws.com/homepage.html
```

La demande se présente comme suit :

```
GET /homepage.html HTTP/1.1
Host: DOC-EXAMPLE-BUCKET1.s3.eu-west-1.amazonaws.com
```

Example – Méthode d'alias CNAME

Pour utiliser cette méthode, vous devez configurer votre nom DNS en tant qu'alias CNAME pour *`bucket-name`*`.s3.us-east-1.amazonaws.com`. Pour de plus amples informations, veuillez consulter Personnalisation des URL Amazon S3 avec des enregistrements CNAME.

Cet exemple utilise les éléments suivants :

- Nom de compartiment - `example.com`
- Nom de clé : `homepage.html`

L'URL se présente comme suit :

```
http://www.example.com/homepage.html
```

L'exemple se présente comme suit :

```
GET /homepage.html HTTP/1.1
Host: www.example.com
```

Personnalisation des URL Amazon S3 avec des enregistrements CNAME

Selon vos besoins, vous préférerez peut-être que `s3.region-code.amazonaws.com` n'apparaisse pas sur votre site web ou service. Par exemple, si vous hébergez des images de site web sur Amazon S3, vous pourriez préférer `http://images.example.com/` à `http://images.example.com.s3.us-east-1.amazonaws.com/`. Tout compartiment avec un nom compatible DNS peut être référencé comme suit : `http://BucketName.s3.Region.amazonaws.com/[Filename]`, par exemple, `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg`. Grâce au CNAME, vous pouvez mapper `images.example.com` à un nom d'hôte Amazon S3 pour que l'URL précédente puisse devenir `http://images.example.com/mydog.jpg`.

Votre nom de compartiment doit être identique à CNAME. Par exemple, si vous créez un CNAME pour mapper `images.example.com` à `images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com/filename` et `http://images.example.com.s3.us-east-1.amazonaws.com/filename` seront identiques.

L'enregistrement du DNS CNAME doit donner au nom de domaine comme alias le nom d'hôte de type hébergement virtuel approprié. Par exemple, si le nom du compartiment et le nom de domaine sont `images.example.com` et que votre compartiment se trouve dans la Région USA Est (Virginie du Nord), l'enregistrement CNAME doit avoir comme alias `images.example.com.s3.us-east-1.amazonaws.com`.

```
images.example.com CNAME    images.example.com.s3.us-east-1.amazonaws.com.
```

Amazon S3 utilise le nom d'hôte pour déterminer le nom du compartiment. C'est pourquoi le CNAME et le nom du compartiment doivent être identiques. Par exemple, admettons que vous avez configuré `www.example.com` comme CNAME pour `www.example.com.s3.us-east-1.amazonaws.com`. Lorsque vous accédez à `http://www.example.com`, Amazon S3 reçoit une demande semblable à la suivante :

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 voit uniquement le nom d'hôte d'origine `www.example.com` et n'a pas connaissance du mappage CNAME utilisé pour résoudre la demande.

Vous pouvez utiliser n'importe quel point de terminaison Amazon S3 dans un alias CNAME. Par exemple, `s3.ap-southeast-1.amazonaws.com` peut être utilisé dans les alias CNAME. Pour plus d'informations sur les points de terminaison, consultez Points de terminaison de demande. Pour créer un site web statique à l'aide d'un domaine personnalisé, consultez Tutoriel : configuration d'un site Web statique à l'aide d'un domaine personnalisé enregistré auprès de Route 53.

> **Important**
>
> Lorsque vous utilisez des URL personnalisées avec des enregistrements CNAME, vous devez vérifier qu'il existe un compartiment correspondant pour tout enregistrement CNAME ou d'alias que vous configurez. Par exemple, si vous créez des entrées DNS pour `www.example.com` et `login.example.com` afin de publier du contenu Web à l'aide de S3, vous devez créer les deux compartiments `www.example.com` et `login.example.com`. Lorsqu'un enregistrement CNAME ou d'alias configuré pointe vers un point de terminaison S3 sans compartiment correspondant, n'importe quel utilisateur AWS peut créer ce compartiment et publier du contenu sous l'alias configuré, même si la propriété n'est pas la même. Pour la même raison, nous vous recommandons de modifier ou de supprimer l'enregistrement CNAME ou d'alias correspondant lors de la suppression d'un compartiment.

Comment associer un nom d'hôte à un compartiment Amazon S3

Pour associer un nom d'hôte à un compartiment Amazon S3 à l'aide d'un alias CNAME

1. Sélectionnez un nom d'hôte qui appartient à un domaine que vous contrôlez.

   Cet exemple utilise le sous-domaine `images` du domaine `example.com`.
2. Créez un compartiment qui correspond au nom d'hôte.

   Dans cet exemple, les noms d'hôte et de compartiment sont `images.example.com`. Le nom du compartiment doit correspondre exactement au nom d'hôte.
3. Créez un enregistrement DNS CNAME qui définit le nom d'hôte comme alias du compartiment Amazon S3.

   Par exemple :

```
images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
```

> **Important**
>
> Pour des raisons d'acheminement de demande, l'enregistrement DNS CNAME doit être défini exactement comme illustré dans l'exemple précédent. Sinon, il peut sembler fonctionner correctement, mais il entraînera à terme un comportement imprévisible.

La procédure de configuration d'enregistrements DNS CNAME dépend de votre serveur ou fournisseur DNS. Pour obtenir des informations spécifiques, consultez la documentation du serveur ou contactez le fournisseur.

Limites

La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Rétrocompatibilité

Les sections suivantes couvrent divers aspects de la rétrocompatibilité Amazon S3 en ce qui concerne les demandes d'URL de type chemin d'accès et hébergement virtuel.

Points de terminaison hérités

Certaines Régions prennent en charge les points de terminaison hérités. Vous pouvez voir ces points de terminaison dans les journaux d'accès au serveur ou les journaux AWS CloudTrail. Pour de plus amples informations, veuillez consulter les informations suivantes. Pour obtenir la liste complète des régions et points de terminaison Amazon S3, consultez Points de terminaison et quotas Amazon S3 dans le Référence générale d'Amazon Web Services.

> **Important**
>
> Même si vous pouvez voir des points de terminaison hérités dans vos journaux, nous vous recommandons de toujours utiliser la syntaxe de point de terminaison standard pour accéder à vos compartiments.
> Les URL de type hébergement virtuel Amazon S3 utilisent le format suivant :

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

Dans Amazon S3, les URL de type chemin d'accès utilisent le format suivant :

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

S3-Région

Certaines régions Amazon S3 plus anciennes prennent en charge des points de terminaison contenant un tiret (`-`) entre `s3` et le code de région (par exemple, `s3-us-west-2`) au lieu d'un point (par exemple, `s3.us-west-2`). Si votre compartiment se trouve dans l'une de ces Régions, vous pouvez voir le format de point de terminaison suivant dans les journaux d'accès au serveur ou les journaux CloudTrail :

```
https://bucket-name.s3-region-code.amazonaws.com
```

Dans cet exemple, le nom du compartiment est DOC-EXAMPLE-BUCKET1 et la région est USA Ouest (Oregon) :

```
https://DOC-EXAMPLE-BUCKET1.s3-us-west-2.amazonaws.com
```

Point de terminaison global hérité

Pour certaines régions, vous pouvez utiliser le point de terminaison global hérité pour élaborer des demandes qui ne spécifient pas de point de terminaison spécifique à la région. Le point de terminaison global hérité est comme suit :

```
bucket-name.s3.amazonaws.com
```

Dans les journaux d'accès au serveur ou les journaux CloudTrail, vous pouvez voir des demandes qui utilisent le point de terminaison global hérité. Dans cet exemple, le nom du compartiment est `DOC-EXAMPLE-BUCKET1` et le point de terminaison global hérité est :

```
https://DOC-EXAMPLE-BUCKET1.s3.amazonaws.com
```

Demandes de type hébergement virtuel pour USA Est (Virginie du Nord)

Les demandes effectuées avec le point de terminaison global hérité sont transmises par défaut à la région USA Est (Virginie du Nord). Par conséquent, le point de terminaison global hérité est parfois utilisé à la place du point de terminaison Régional pour USA Est (Virginie du Nord). Si vous créez un compartiment dans la Région USA Est (Virginie du Nord) et que vous utilisez le point de terminaison global, Amazon S3 achemine votre demande vers cette Région par défaut.

Demandes de type hébergement virtuel pour d'autres régions

Le point de terminaison global hérité est également utilisé pour les demandes de type hébergement virtuel dans d'autres régions prises en charge. Si vous créez un compartiment dans une Région qui a été lancée avant le 20 mars 2019 et que vous utilisez le point de terminaison global hérité, Amazon S3 met à jour l'enregistrement DNS pour rediriger la demande vers l'emplacement correct, ce qui peut prendre du temps. Entre-temps, la règle par défaut s'applique et la demande de type hébergement virtuel est acheminée vers la région USA Est (Virginie du Nord). Amazon S3 la redirige ensuite avec une redirection temporaire HTTP 307 vers la région appropriée.

Pour les compartiments S3 dans les régions lancées après le 20 mars 2019, le serveur DNS n'achemine pas votre demande directement vers la Région AWS dans laquelle se trouve votre compartiment. Au lieu de cela, il renvoie une erreur HTTP 400 Requête erronée. Pour de plus amples informations, veuillez consulter Demandes.

Demandes de type chemin d'accès

Pour la région USA Est (Virginie du Nord), vous pouvez utiliser le point de terminaison global hérité pour les demandes de type chemin d'accès.

Pour toutes les autres Régions, la syntaxe de type chemin exige l'utilisation du point de terminaison spécifique à la Région lors d'une tentative d'accès à un compartiment. Si vous essayez d'accéder à un compartiment avec le point de terminaison global hérité ou un autre point de terminaison qui est différent de celui de la Région où le compartiment réside, vous recevez un code de réponse HTTP 307 d'erreur de redirection temporaire et un message indiquant l'URI correct pour votre ressource. Par exemple, si vous utilisez `https://s3.amazonaws.com/`*`bucket-name`* pour un compartiment qui a été créé dans la Région USA Ouest (Oregon), vous recevez une erreur HTTP 307 Temporary Redirect (Redirection temporaire).

Redirection de demande et API REST

Rubriques

- Redirections et agents utilisateurs HTTP

Cette section décrit comment traiter les redirections HTTP à l'aide de l'API REST Amazon S3. Pour obtenir des informations générales sur les redirections Amazon S3, veuillez consulter Demandes dans la Référence d'API Amazon Simple Storage Service.

Redirections et agents utilisateurs HTTP

Les programmes qui utilisent l'API REST Amazon S3 doivent traiter les redirections au niveau de la couche applicative ou de la couche HTTP. De nombreux agents utilisateurs et bibliothèques client HTTP peuvent être configurés pour traiter automatiquement correctement les redirections. Toutefois, de nombreux autres présentent des implémentations de redirection incorrectes ou incomplètes.

Avant de faire confiance à une bibliothèque pour s'acquitter de l'obligation de redirection, testez les cas suivants :

- Vérifiez que tous les en-têtes de demande HTTP sont correctement inclus dans la demande redirigée (la seconde demande après réception d'une redirection), y compris les standards HTTP tels que Authorization et Date.
- Vérifiez que les redirections autres que GET, telles que PUT et DELETE, fonctionnent correctement.
- Vérifiez que les demandes PUT de grande taille suivent correctement les redirections.
- Vérifiez que les demandes PUT suivent correctement les redirections si la réponse 100-continue met longtemps à arriver.

Les agents utilisateurs HTTP qui respectent scrupuleusement la RFC 2616 peuvent nécessiter une confirmation explicite avant de suivre une redirection lorsque la méthode de demande HTTP n'est pas GET ni HEAD. Il est généralement sans danger de suivre les redirections générées automatiquement par Amazon S3, car le système émet des redirections uniquement vers les hôtes situés au sein du domaine amazonaws.com et l'effet de la demande redirigée est identique à celui de la demande d'origine.

Redirections et 100-continue

Pour simplifier le traitement de la redirection, améliorer l'efficacité et éviter les coûts associés à l'envoi répété du corps d'une demande redirigée, configurez l'application pour qu'elle utilise des expressions

100-continue pour les opérations PUT. Lorsque votre application utilise 100-continue, elle n'envoie pas le corps de la demande tant qu'elle ne reçoit pas d'accusé de réception. Si le message est rejeté sur la base des en-têtes, le corps du message n'est pas envoyé. Pour plus d'informations sur 100-continue, consultez la RFC 2616, section 8.2.3

> **Note**
>
> Selon la RFC 2616, lorsque vous utilisez `Expect: Continue` avec un serveur HTTP inconnu, vous ne devez pas attendre un temps indéfini avant d'envoyer le corps de la demande. Cela est dû au fait que certains serveurs HTTP ne reconnaissent pas l'expression 100-continue. Toutefois, Amazon S3 reconnaît que la demande contient une expression `Expect: Continue` et répond avec un statut 100-continue provisoire ou un code de statut final. En outre, aucune erreur de redirection ne surviendra après la réception de l'autorisation provisoire 100-continue. Cela vous aidera à éviter de recevoir une réponse de redirection alors que vous écrivez encore le corps de la demande.

Exemple de redirection

Cette section fournit un exemple d'interaction client-serveur utilisant des redirections HTTP et 100-continue.

Voici un exemple de commande PUT pour le compartiment `quotes.s3.amazonaws.com`.

```
PUT /nelson.txt HTTP/1.1
Host: quotes.s3.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

Amazon S3 retourne les informations suivantes :

```
HTTP/1.1 307 Temporary Redirect
Location: http://quotes.s3-4c25d83b.amazonaws.com/nelson.txt?rk=8d47490b
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Mon, 15 Oct 2007 22:18:46 GMT
```

```
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>TemporaryRedirect</Code>
  <Message>Please re-send this request to the
  specified temporary endpoint. Continue to use the
  original request endpoint for future requests.
  </Message>
  <Endpoint>quotes.s3-4c25d83b.amazonaws.com</Endpoint>
  <Bucket>quotes</Bucket>
</Error>
```

Le client suit la réponse de redirection et émet une nouvelle demande adressée au point de terminaison temporaire `quotes.s3-4c25d83b.amazonaws.com`.

```
PUT /nelson.txt?rk=8d47490b HTTP/1.1
Host: quotes.s3-4c25d83b.amazonaws.com
Date: Mon, 15 Oct 2007 22:18:46 +0000

Content-Length: 6
Expect: 100-continue
```

Amazon S3 retourne une expression 100-continue indiquant que le client doit à présent envoyer le corps de la demande.

```
HTTP/1.1 100 Continue
```

Le client envoie le corps de la demande.

```
ha ha\n
```

Amazon S3 retourne la réponse finale.

```
HTTP/1.1 200 OK
Date: Mon, 15 Oct 2007 22:18:48 GMT

ETag: "a2c8d6b872054293afd41061e93bc289"
Content-Length: 0
Server: AmazonS3
```

Développement avec Amazon S3 à l'aide de la AWS CLI

Suivez ces étapes pour télécharger et configurer l'AWS Command Line Interface (AWS CLI).

Pour obtenir la liste des commandes de la AWS CLI pour Amazon S3, consultez les pages suivantes dans la Référence des commandes de la AWS CLI :

- s3
- s3api
- s3control

> **Note**
>
> Les services dans AWS, tels que Amazon S3, nécessitent que vous fournissiez vos informations d'identification lors de l'accès. Le service peut ainsi déterminer si vous avez les autorisations pour accéder à ses ressources. La console exige votre mot de passe. Vous pouvez créer des clés d'accès pour votre Compte AWS afin d'accéder à la AWS CLI ou à l'API. Cependant, nous vous déconseillons d'accéder à AWS avec les informations d'identification de votre Compte AWS. À la place, nous vous recommandons d'utiliser AWS Identity and Access Management (). Créez un utilisateur IAM, ajoutez-le à un groupe IAM avec des autorisations administratives, puis attribuez-lui ces autorisations. Vous pouvez alors accéder à AWS à l'aide d'une URL spéciale et des informations d'identification de cet utilisateur IAM. Pour obtenir des instructions, veuillez accéder à Création de votre premier groupe d'utilisateurs et d'administrateurs IAM dans le Guide de l'utilisateur IAM.

Pour configurer l AWS CLI

1. Téléchargez et configurez l'interface AWS CLI. Pour obtenir des instructions, consultez les rubriques suivantes dans le Guide de l'utilisateur de l'interface AWS Command Line Interface :
   - Préparation de l'installation de l'AWS Command Line Interface
   - Configuration de l'AWS Command Line Interface (français non garanti)
2. Ajoutez un profil désigné pour l'utilisateur administrateur dans le fichier de configuration de l'AWS CLI. Vous utiliserez ce profil lorsque vous exécuterez les commandesAWS CLI. Pour plus d'informations, consultez Profils nommés pour AWS CLI dans le Guide de l'utilisateur AWS Command Line Interface.

```
[adminuser]
aws_access_key_id = adminuser access key ID
aws_secret_access_key = adminuser secret access key
region = aws-region
```

Pour obtenir la liste des Régions AWS disponibles, consultez Régions et points de terminaison dans la Références générales AWS.

3. Vérifiez la configuration en saisissant les commandes suivantes à l'invite de commande.

   - Essayez la commande `help` pour vérifier que l'AWS CLI est installée sur votre ordinateur :

     ```
     aws help
     ```

   - Exécutez une commande `S3` à l'aide des informations d'identification `adminuser` que vous venez de créer. Pour ce faire, ajoutez le paramètre `--profile` à votre commande pour spécifier le nom du profil. Dans cet exemple, la commande `ls` répertorie les compartiments inclus dans votre compte. L'AWS CLI utilise les informations d'identification `adminuser` pour authentifier la demande.

     ```
     aws s3 ls --profile adminuser
     ```

Développement avec Amazon S3 à l'aide des AWS SDK

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

> **Note**
>
> Vous pouvez l'utiliser AWS Amplify pour le end-to-end développement complet d'applications Web et mobiles. Amplify Storage intègre parfaitement les fonctionnalités de stockage et de gestion de fichiers dans les applications Web et mobiles frontales, basées sur Amazon S3. Pour plus d'informations, consultez la section Stockage dans le guide de l'utilisateur d'Amplify.

Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDK) sont disponibles pour de nombreux langages de programmation populaires. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Pour voir des exemples spécifiques à ce service, consultez Exemples de code pour Amazon S3 à l'aide de AWS kits SDK.

Exemple de disponibilité

Vous n’avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Faire un commentaire en bas de cette page.

Interfaces de programmation du SDK

Chaque AWS SDK fournit une ou plusieurs interfaces de programmation pour travailler avec Amazon S3. Chaque SDK fournit une interface de bas niveau pour Amazon S3, avec des méthodes qui ressemblent beaucoup aux opérations d'API. Certains SDK fournissent des interfaces de haut niveau pour Amazon S3, qui sont des abstractions destinées à simplifier les cas d'utilisation courants.

Par exemple, lorsque vous effectuez un téléchargement partitionné à l'aide des opérations d'API de bas niveau, vous devez utiliser une opération pour lancer le téléchargement, une autre pour télécharger des parties et une opération finale pour terminer le téléchargement. Une opération d'API de téléchargement en plusieurs parties de haut niveau vous permet d'effectuer toutes les opérations requises pour le téléchargement en un seul appel d'API. Pour obtenir des exemples, consultez Chargement d'un objet à l'aide du chargement partitionné.

Les opérations d'API de bas niveau permettent de mieux contrôler le téléchargement. Nous vous recommandons d'utiliser les opérations d'API de bas niveau si vous devez suspendre et reprendre les téléchargements, modifier la taille des pièces pendant le téléchargement ou commencer les téléchargements lorsque vous ne connaissez pas la taille des données à l'avance.

Spécification de la version de signature dans l'authentification de la demande

Amazon S3 ne prend en charge que AWS la version 4 de Signature dans la plupart des cas Régions AWS. Dans certains des modèles les plus anciens Régions AWS, Amazon S3 prend en charge à la fois la version 4 et la version 2 de Signature. Cependant, Signature Version 2 est en cours de désactivation (obsolète). Pour de plus amples informations sur la fin de la prise en charge de Signature Version 2, veuillez consulter AWS Signature version 2 désactivée (obsolète) pour Amazon S3.

Pour obtenir la liste de toutes les régions Amazon S3 et savoir quelles sont les versions de signatures qu'elles prennent en charge, veuillez consulter Régions et points de terminaisons dans les Références générales AWS .

Pour tous Régions AWS, les AWS SDK utilisent Signature Version 4 par défaut pour authentifier les demandes. Lorsque vous utilisez AWS des SDK publiés avant mai 2016, vous devrez peut-être demander la version 4 de Signature, comme indiqué dans le tableau suivant.

Kit SDK	Demande de la Signature Version 4 pour l'authentification des demandes
AWS CLI	Pour le profil par défaut, exécutez la commande suivante : `$ aws configure set default.s3.signature_version s3v4` Pour un profil personnalisé, exécutez la commande suivante : `$ aws configure set profile.your_profile_name.s3.signature_version s3v4`
Kit SDK Java	Ajoutez la chaîne suivante à votre code : `System.setProperty(SDKGlobalConfiguration.ENABLE_S3_SIGV4_SYSTEM_PROPERTY, "true");` Sinon, vous pouvez spécifier la commande ci-dessous dans la ligne de commande : `-Dcom.amazonaws.services.s3.enableV4`
JavaScript SDK	Définissez le paramètre `signatureVersion` sur `v4` lors de la création du client: `var s3 = new AWS.S3({signatureVersion: 'v4'});`
Kit SDK PHP	Définissez le paramètre `signature` sur `v4` lors de la création du client de service Amazon S3 pour le kit SDK PHP v2 : `<?php` `$client = S3Client::factory([` `'region' => 'YOUR-REGION',`

Kit SDK	Demande de la Signature Version 4 pour l'authentification des demandes
	<pre> 'version' => 'latest', 'signature' => 'v4']);</pre> Lorsque vous utilisez le kit SDK PHP v3, définissez le paramètre `signature_version` sur `v4` lors de la création du client de service Amazon S3 : <pre><?php $s3 = new Aws\S3\S3Client(['version' => '2006-03-01', 'region' => 'YOUR-REGION', 'signature_version' => 'v4']);</pre>
Kit SDK Python-Boto	Spécifiez la chaîne suivante dans le fichier de configuration par défaut boto: <pre>[s3] use-sigv4 = True</pre>
Kit SDK Ruby	Kit SDK Ruby – Version 1 : définissez le paramètre `:s3_signature_version` sur `:v4` lors de la création du client: <pre>s3 = AWS::S3::Client.new(:s3_signature_version => :v4)</pre> Kit SDK Ruby – Version 3 : définissez le paramètre `signature_version` sur `v4` lors de la création du client: <pre>s3 = Aws::S3::Client.new(signature_version: 'v4')</pre>

Kit SDK	Demande de la Signature Version 4 pour l'authentification des demandes
Kit SDK .NET	Ajoutez la chaîne suivante au code avant de créer le client Amazon S3 : `AWSConfigsS3.UseSignatureVersion4 = true;` Sinon, vous pouvez ajouter le code ci-dessous au fichier de configuration : `<appSettings>` `  <add key="AWS.S3.UseSignatureVersion4" value="tr` `ue" />` `</appSettings>`

AWS Signature version 2 désactivée (obsolète) pour Amazon S3

Signature Version 2 est en cours de désactivation (obsolète) dans Amazon S3. Une fois cette version devenue obsolète, Amazon S3 n'acceptera plus que les demandes d'API signées avec Signature Version 4.

Cette section fournit des réponses aux questions fréquentes concernant la fin de la prise en charge de Signature Version 2.

Qu'est-ce que Signature Version 2/4 et qu'implique la signature des demandes ?

Le processus de signature de Signature Version 2 ou Signature Version 4 est utilisé pour authentifier vos demandes d'API Amazon S3. La signature des demandes permet à Amazon S3 d'identifier l'auteur de l'envoi de la demande et protège vos demandes contre les utilisateurs malveillants.

Pour plus d'informations sur les AWS demandes de signature, consultez la section Signature des demandes d' AWS API dans le Références générales AWS.

En quoi consiste la mise à jour ?

Nous prenons actuellement en charge les demandes d'API Amazon S3 qui sont signées avec les processus Signature Version 2 et Signature Version 4. Une fois la version 2 devenue obsolète, Amazon S3 n'acceptera plus que les demandes signées avec Signature Version 4.

Pour plus d'informations sur les AWS demandes de signature, consultez la section Modifications apportées à la version 4 de la signature dans le Références générales AWS.

Quel est l'objectif de cette mise à jour ?

Signature Version 4 améliore la sécurité grâce à l'utilisation d'une clé de signature au lieu d'une clé d'accès secrète. La version 4 de Signature est actuellement prise en charge dans tous les pays Régions AWS, tandis que la version 2 de Signature n'est prise en charge que dans les régions lancées avant janvier 2014. Cette mise à jour nous permet d'offrir une expérience plus homogène dans toutes les régions.

Comment vérifier que j'utilise Signature Version 4 et quelles sont les mises à jour dont j'ai besoin ?

La version de signature utilisée pour signer les demandes est généralement définie par l'outil ou le kit SDK du côté du client. Par défaut, les dernières versions de nos AWS SDK utilisent Signature Version 4. Pour les logiciels tiers, contactez l'équipe d'assistance appropriée de votre logiciel pour confirmer la version dont vous avez besoin. Si vous envoyez des appels REST directs à Amazon S3, vous devez modifier votre application pour qu'elle utilise le processus de signature Signature Version 4.

Pour plus d'informations sur la version des AWS SDK à utiliser lors du passage à la version 4 de Signature, consultezPassage de Signature Version 2 à Signature Version 4.

Pour de plus amples informations sur l'utilisation de Signature Version 4 avec l'API REST Amazon S3, veuillez consulter Authentification des demandes (AWS Signature Version 4) dans la Référence API Amazon Simple Storage Service.

Que se passe-t-il si je n'effectue pas les mises à niveau ?

L'authentification des demandes signées avec Signature Version 2 échouera avec Amazon S3. Les demandeurs verront des erreurs indiquant que la demande doit être signée avec Signature Version 4.

Dois-je effectuer des modifications même si j'utilise une URL présignée qui exige que je signe pour plus de 7 jours ?

Si vous utilisez une URL présignée qui exige que vous signiez pour plus de 7 jours, aucune action n'est requise de votre part. Vous pouvez continuer à utiliser AWS la version 2 de Signature

pour signer et authentifier l'URL présignée. Nous fournirons prochainement des informations supplémentaires sur le passage à Signature Version 4 avec les URL présignées.

Plus d'informations

- Pour plus d'informations sur l'utilisation de Signature version 4, consultez Signing AWS API Requests.
- Consultez la liste des modifications entre Signature Version 2 et Signature Version 4 dans Modification dans Signature Version 4.
- Consultez l'article AWS Signature Version 4 pour remplacer AWS Signature Version 2 pour signer les demandes d'API Amazon S3 dans les AWS forums.
- Si vous avez des questions, veuillez contacter le AWS Support.

Passage de Signature Version 2 à Signature Version 4

Si vous utilisez actuellement Signature Version 2 pour authentifier vos demandes d'API Amazon S3, vous devez passer à Signature Version 4. Signature Version 2 ne sera bientôt plus pris en charge, comme décrit dans AWS Signature version 2 désactivée (obsolète) pour Amazon S3.

Pour de plus amples informations sur l'utilisation de Signature Version 4 avec l'API REST Amazon S3, veuillez consulter Authentification des demandes (AWS Signature Version 4) dans la Référence API Amazon Simple Storage Service.

La liste suivante répertorie les kits SDK avec la version minimum nécessaire pour utiliser Signature Version 4 (SigV4). Si vous utilisez des URL présignées avec les SDK AWS Java, JavaScript (Node.js) ou Python (Boto/CLI), vous devez définir la version correcte Région AWS et définir la version 4 de signature dans la configuration du client. Pour de plus amples informations sur la définition de `SigV4` dans la configuration client, veuillez consulter Spécification de la version de signature dans l'authentification de la demande.

Si vous utilisez ce kit SDK/produit	Passez à cette version de SDK	Modification de code requise dans le client pour utiliser Sigv4 ?	Lien vers la documentation SDK
AWS SDK for Java v1	Passez à Java 1.11.201+ ou v2.	Oui	Spécification de la version de signature dans l'authentification de la demande
AWS SDK for Java v2	Pas de mise à niveau de SDK nécessaire.	Non	AWS SDK for Java
AWS SDK for .NET v1	Passez à 3.1.10 ou version ultérieure.	Oui	AWS SDK for .NET
AWS SDK for .NET v2	Passez à 3.1.10 ou version ultérieure.	Non	AWS SDK for .NET v2
AWS SDK for .NET v3	Passez à 3.3.0.0 ou version ultérieure.	Oui	AWS SDK for .NET v3
AWS SDK for JavaScript v1	Passez à 2.68.0 ou version ultérieure.	Oui	AWS SDK for JavaScript

Si vous utilisez ce kit SDK/produit	Passez à cette version de SDK	Modification de code requise dans le client pour utiliser Sigv4 ?	Lien vers la documentation SDK
AWS SDK for JavaScript v2	Passez à 2.68.0 ou version ultérieure.	Oui	AWS SDK for JavaScript
AWS SDK for JavaScript v3	Aucune autre action n'est requise. Passez à la version majeure V3 au cours du 3e trimestre 2019.	Non	AWS SDK for JavaScript
AWS SDK for PHP v1	Recommandation de mise à niveau vers la version la plus récente de PHP ou, au moins, vers v2.7.4 avec le paramètre de signature défini sur v4 dans la configuration du client S3.	Oui	AWS SDK for PHP

Si vous utilisez ce kit SDK/produit	Passez à cette version de SDK	Modification de code requise dans le client pour utiliser Sigv4 ?	Lien vers la documentation SDK
AWS SDK for PHP v2	Recommandation de mise à niveau vers la version la plus récente de PHP ou, au moins, vers v2.7.4 avec le paramètre de signature défini sur v4 dans la configuration du client S3.	Non	AWS SDK for PHP
AWS SDK for PHP v3	Pas de mise à niveau de SDK nécessaire.	Non	AWS SDK for PHP
Boto2	Passez à Boto2 v2.49.0.	Oui	Mise à niveau Boto 2
Boto3	Passez à 1.5.71 (Botocore), 1.4.6 (Boto3).	Oui	Boto 3 - AWS SDK pour Python

Si vous utilisez ce kit SDK/produit	Passez à cette version de SDK	Modification de code requise dans le client pour utiliser Sigv4 ?	Lien vers la documentation SDK
AWS CLI	Passez à 1.11.108.	Oui	AWS Command Line Interface
AWS CLI v2 (aperçu)	Pas de mise à niveau de SDK nécessaire.	Non	AWS Command Line Interface version 2
AWS SDK for Ruby v1	Passez à Ruby V3.	Oui	Ruby V3 pour AWS
AWS SDK for Ruby v2	Passez à Ruby V3.	Oui	Ruby V3 pour AWS
AWS SDK for Ruby v3	Pas de mise à niveau de SDK nécessaire.	Non	Ruby V3 pour AWS
Go	Pas de mise à niveau de SDK nécessaire.	Non	AWS SDK for Go
C++	Pas de mise à niveau de SDK nécessaire.	Non	AWS SDK for C++

AWS Tools for Windows PowerShell ou AWS Tools for PowerShell Core

Si vous utilisez des versions de module antérieures à 3.3.0.0, vous devez passer à 3.3.0.0.

Pour obtenir les informations de version, utilisez le cmdlet `Get-Module` :

```
Get-Module –Name AWSPowershell
Get-Module –Name AWSPowershell.NetCore
```

Pour mettre à jour la version 3.3.0.0, utilisez le cmdlet `Update-Module` :

```
Update-Module –Name AWSPowershell
Update-Module –Name AWSPowershell.NetCore
```

Vous pouvez utiliser des URL présignées valides pendant plus de 7 jours sur lesquelles vous pouvez envoyer le trafic Signature Version 2.

Développer avec Amazon S3 à l’aide de l’API REST

L’architecture d’Amazon S3 a été conçue de manière à être neutre en termes de langage de programmation et utilise nos interfaces supportées pour stocker et récupérer des objets.

Amazon S3 fournit actuellement une interface REST. Avec REST, les métadonnées sont renvoyées dans les en-têtes HTTP. Etant donné que nous prenons uniquement en charge les demandes HTTP de 4 Ko maximum (excluant le corps), la quantité de métadonnées que vous pouvez envoyer est limitée. L'API REST est une interface HTTP pour Amazon S3. Lorsque vous utilisez REST, vous utilisez des demandes HTTP standard pour créer, récupérer et supprimer des compartiments et des objets.

Vous pouvez choisir n'importe quelle boîte à outils prenant en charge HTTP pour utiliser l'API REST. Vous pouvez même utiliser un explorateur pour récupérer des objets, si ceux-ci peuvent être lus de manière anonyme.

L'API REST utilise les codes de statut et en-têtes HTTP standard afin de permettre aux explorateurs et boîtes à outils classiques de fonctionner. Dans certains zones, nous avons ajouté des fonctionnalités à HTTP (par exemple, nous avons ajouté des en-têtes afin de permettre le

contrôle d'accès). Dans ces cas précis, nous avons fait notre possible pour intégrer cette nouvelle fonctionnalité de sorte qu'elle corresponde à la manière dont HTTP est généralement utilisé.

Pour plus d'informations sur l'envoi de demandes à l'aide de l'API REST, consultez Demandes à l'aide de l'API REST. Pour en savoir plus sur les éléments à ne pas oublier lors de l'utilisation de l'API REST, consultez les rubriques ci-dessous.

Pour plus d'informations sur l'API REST Amazon S3, veuillez consulter la Référence d'API Amazon Simple Storage Service.

Rubriques

Demande de routage

Les programmes qui émettent des demandes sur les compartiments créés à l'aide de l'API CreateBucket, qui incluent CreateBucketConfiguration, doivent prendre en charge les redirections. De plus, certains clients qui ne respectent pas la durée de vie des demandes DNS risquent de rencontrer des problèmes.

Cette section décrit les demandes de routage et les erreurs DNS à prendre en compte lors de la conception de votre service ou de votre application à utiliser avec Amazon S3.

Redirection de demande et API REST

Amazon S3 utilise le système de noms de domaine (DNS) pour diriger les demandes vers les installations susceptibles de les traiter. Ce système fonctionne de façon efficace, mais des erreurs de routage temporaires peuvent survenir. Si une demande arrive à une destination Amazon S3 erronée, Amazon S3 répond par une redirection temporaire qui indique au demandeur de renvoyer la demande à un nouveau point de terminaison. Si une demande est mal formulée, Amazon S3 utilise des redirections permanentes pour fournir des instructions sur la manière de formuler correctement la demande.

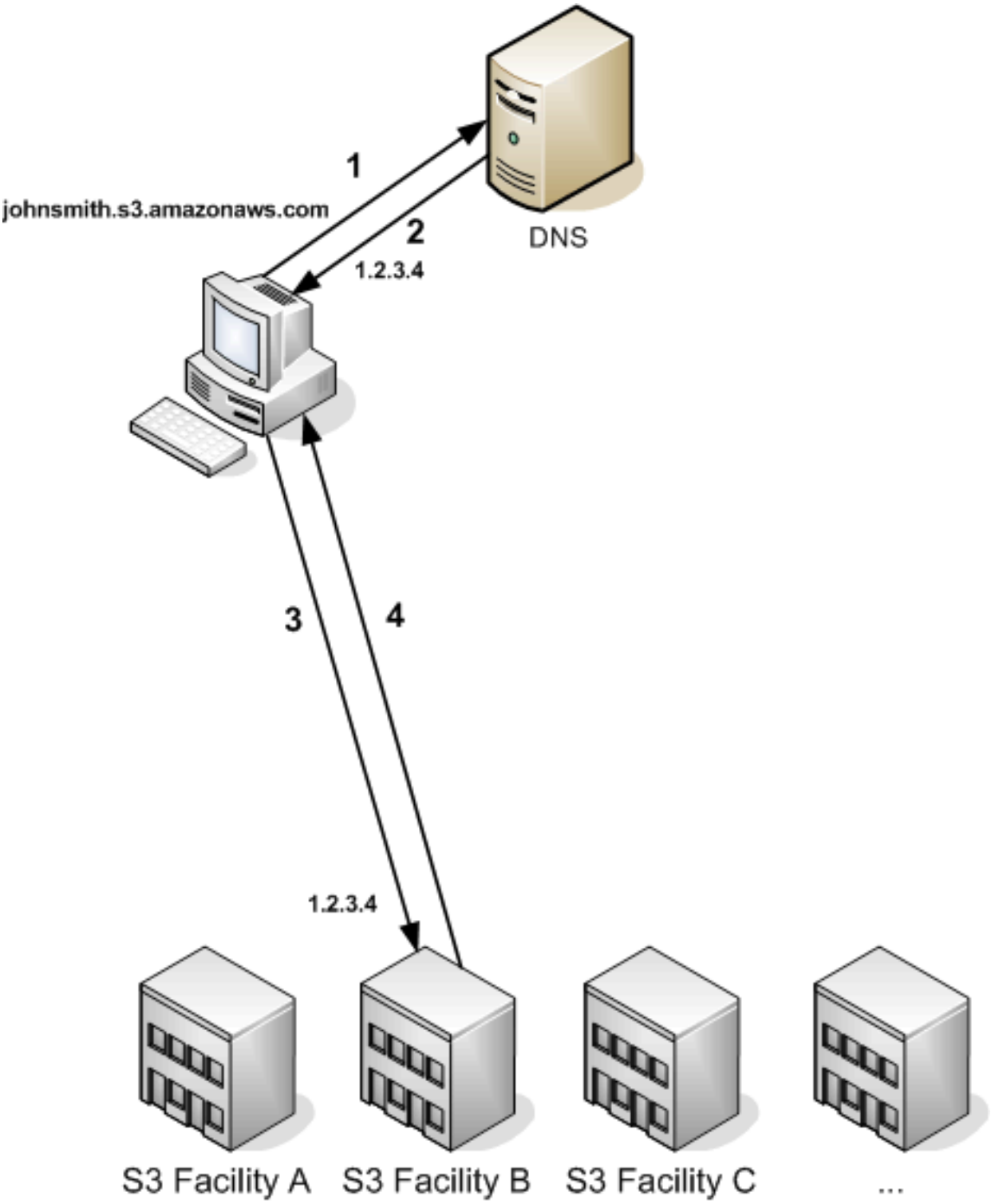
> **Important**
>
> Pour utiliser cette fonction, vous devez disposer d'une application capable de gérer les réponses de redirection Amazon S3. La seule exception concerne les applications qui fonctionnent avec des compartiments qui ont été créés sans

`<CreateBucketConfiguration>`. Pour plus d'informations sur les contraintes d'emplacement, consultez Accès à un compartiment Amazon S3 et liste des compartiments.
Pour toutes les Régions lancées après le 20 mars 2019, si une demande arrive à l'emplacement Amazon S3 incorrect, Amazon S3 renvoie une erreur HTTP 400 Requête incorrecte.
Pour plus d'informations sur l'activation et la désactivation d'une Région AWS, consultez Régions AWS et points de terminaison dans la Références générales AWS.

Rubriques

routage DNS

Le routage DNS achemine les demandes vers les installations Amazon S3 appropriées. La figure et la procédure suivantes montrent un exemple de routage DNS.



Étapes des demandes de routage DNS

1. Le client effectue une demande DNS pour obtenir un objet stocké sur Amazon S3.

2. Le client reçoit une ou plusieurs adresses IP des installations qui peuvent traiter la demande. Dans cet exemple, l'adresse IP est pour l'installation B.
3. Le client effectue une demande adressée à l'installation B Amazon S3.
4. L'installation B renvoie une copie de l'objet au client.

Redirection de demande temporaire

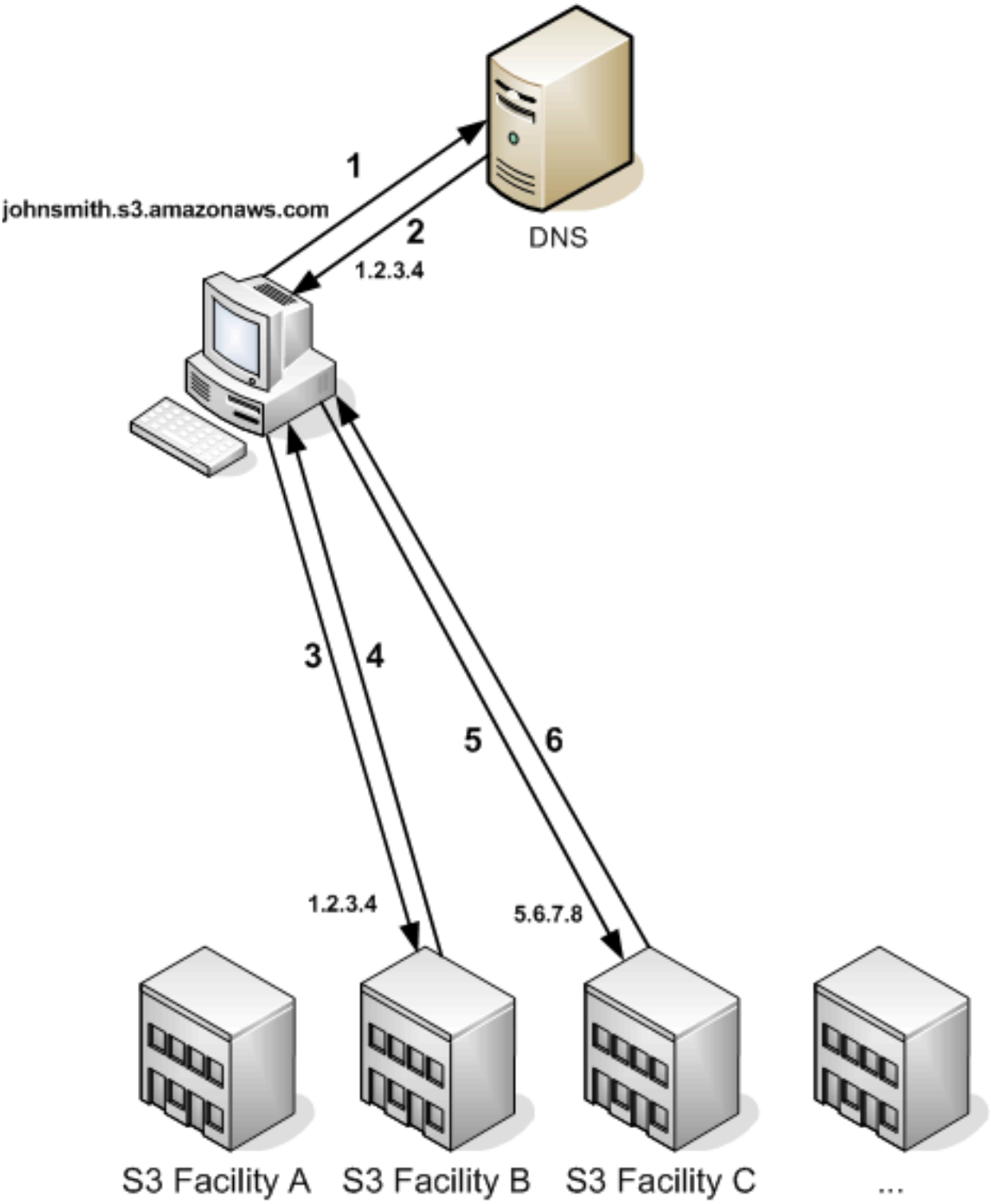
Une redirection temporaire est un type de réponse d'erreur qui indique au demandeur qu'il doit renvoyer la demande à un autre point de terminaison. En raison de la nature distribuée d'Amazon S3, les demandes peuvent être acheminées temporairement vers la mauvaise installation. Cela a plus de chances de se produire immédiatement après la création ou la suppression de compartiments.

Par exemple, si vous créez un compartiment et adressez immédiatement une demande à ce compartiment, vous pouvez recevoir une redirection temporaire, selon la contrainte d'emplacement du compartiment. Si vous avez créé le compartiment dans la Région AWS USA Est (Virginie du Nord), vous ne verrez pas la redirection, car il s'agit également du point de terminaison Amazon S3 par défaut.

Toutefois, si le compartiment est créé dans une autre Région, toutes les demandes adressées au compartiment iront au point de terminaison par défaut lors de la propagation de l'entrée DNS du compartiment. Le point de terminaison par défaut redirige la demande vers le point de terminaison correct avec une réponse HTTP 302. Les redirections temporaires contiennent un URI vers l'installation correcte, que vous pouvez utiliser pour renvoyer immédiatement la demande.

> **Important**
>
> Ne réutilisez pas un point de terminaison fourni par une réponse de redirection précédente. Cela peut sembler fonctionner (même pendant de longues périodes), mais cela peut fournir des résultats imprévisibles, voire entraîner un échec sans préavis.

La figure et la procédure suivantes montrent un exemple de redirection temporaire.



Étapes d'une redirection de demande temporaire

1. Le client effectue une demande DNS pour obtenir un objet stocké sur Amazon S3.
2. Le client reçoit une ou plusieurs adresses IP des installations qui peuvent traiter la demande.

3. Le client effectue une demande adressée à l'installation B Amazon S3.
4. L'installation B renvoi une redirection indiquant que l'objet est disponible à partir de l'emplacement C.
5. Le client renvoie la demande à l'installation C.
6. L'installation C renvoie une copie de l'objet.

Redirection de demande permanente

Une redirection permanente indique que votre demande s'adressait de façon inappropriée à une ressource. Par exemple, des redirections permanentes se produisent si vous utilisez une demande de type chemin pour accéder à un compartiment qui a été créé avec `<CreateBucketConfiguration>`. Pour de plus amples informations, veuillez consulter Accès à un compartiment Amazon S3 et liste des compartiments.

Pour vous aider à détecter ces erreurs au cours du développement, ce type de redirection ne contient pas d'en-tête HTTP d'emplacement vous permettant de suivre automatiquement la demande vers l'emplacement correct. Veuillez consulter le document d'erreur XML obtenu comme aide pour utiliser le point de terminaison Amazon S3 correct.

Exemples de redirection de demande

Voici des exemples de réponses à des redirections de demande temporaires.

Réponse de redirection temporaire d'API REST

```
HTTP/1.1 307 Temporary Redirect
Location: http://awsexamplebucket1.s3-gztb4pa9sq.amazonaws.com/photos/puppy.jpg?
rk=e2c69a31
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Fri, 12 Oct 2007 01:12:56 GMT
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>TemporaryRedirect</Code>
  <Message>Please re-send this request to the specified temporary endpoint.
  Continue to use the original request endpoint for future requests.</Message>
  <Endpoint>awsexamplebucket1.s3-gztb4pa9sq.amazonaws.com</Endpoint>
```

```
</Error>
```

Réponse de redirection temporaire d'API SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

```
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.TemporaryRedirect</Faultcode>
    <Faultstring>Please re-send this request to the specified temporary endpoint.
    Continue to use the original request endpoint for future requests.</Faultstring>
    <Detail>
      <Bucket>images</Bucket>
      <Endpoint>s3-gztb4pa9sq.amazonaws.com</Endpoint>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```

Considérations DNS

Une disponibilité extrêmement élevée est l'un des prérequis pour la conception d'Amazon S3. Afin de répondre à cette exigence, nous devons actualiser les adresses IP associées au point de terminaison Amazon S3 dans le DNS si nécessaire. Ces modifications apparaissent automatiquement sur les clients à court terme, mais pas sur certains clients à long terme. Les clients à long terme devront effectuer des actions spécifiques pour résoudre à nouveau le point de terminaison Amazon S3 périodiquement afin de bénéficier de ces modifications. Pour plus d'informations sur les machines virtuelles, consultez les rubriques suivantes :

- Pour Java, JVM de Sun met toujours en cache les recherches DNS par défaut ; consultez « InetAddress Caching » de la documentation InetAddress pour savoir comment modifier ce comportement.

- Pour PHP, l'ordinateur virtuel persistant qui utilise les configurations de développement les plus couramment utilisées met en cache les recherches jusqu'à ce que l'ordinateur virtuel redémarre. Consultez les documents getHostByName PHP.

Gestion des erreurs REST et SOAP

Rubriques

- Réponse d'erreur REST
- Réponse d'erreur SOAP
- Bonnes pratiques concernant les erreurs Amazon S3

Cette section décrit les erreurs REST et SOAP et la façon de les traiter.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Réponse d'erreur REST

Si une demande REST génère une erreur, la réponse HTTP a :

- un document d'erreur XML comme réponse ;
- un en-tête Content-Type : application/xml ;
- un code de statut HTTP 3xx, 4xx ou 5xx.

Voici un exemple de réponse d'erreur REST.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
```

```
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

Pour de plus amples informations sur les erreurs Amazon S3, veuillez consulter ErrorCodeList.

En-têtes de réponse

Ci-après, des en-têtes de réponse renvoyés par toutes les opérations :

- `x-amz-request-id:` Un ID unique affecté par le système à chaque demande. Dans l'éventualité peu probable où Amazon S3 poserait des problèmes, Amazon peut utiliser cet ID pour résoudre le problème.
- `x-amz-id-2:` Un jeton spécial qui nous aidera à résoudre des problèmes.

Réponse d'erreur

Quand une demande Amazon S3 est erronée, le client reçoit une réponse d'erreur. Le format exact de la réponse d'erreur est spécifique à l'API : à titre d'exemple, la réponse d'erreur REST diffère de la réponse d'erreur SOAP. Néanmoins, toutes les réponses d'erreur ont des éléments communs.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Code d'erreur

Le code d'erreur est une chaîne qui identifie de façon univoque une condition d'erreur. Il doit être lu et compris par des programmes qui détectent et gèrent des erreurs par type. Beaucoup de codes d'erreur sont communs aux API SOAP et REST, mais certains sont spécifiques aux API. A titre d'exemple, le code d'erreur NoSuchKey est universel, mais UnexpectedContent n'est généré qu'à la suite d'une demande REST non valide. Dans tous les cas, les codes d'erreur SOAP ont un préfixe comme indiqué dans le tableau des codes d'erreur, ainsi une erreur NoSuchKey est renvoyée dans SOAP comme Client.NoSuchKey.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Message d'erreur

Le message d'erreur contient une description générique de la condition d'erreur en anglais. Il s'adresse à un public d'individus. Des programmes basiques affichent le message directement pour l'utilisateur final s'ils trouvent une condition d'erreur qu'ils ne savent, ni souhaitent gérer. Des programmes sophistiqués avec une gestion d'erreur plus complète et une stratégie d'internationalisation personnalisée risquent plus d'ignorer le message d'erreur.

Détails complémentaires

Beaucoup de réponses d'erreur contiennent des données structurées complémentaires destinées à être lues et comprises par un développeur diagnostiquant des erreurs de programmation. A titre d'exemple, si vous envoyez un en-tête Content-MD5 avec une demande REST PUT qui ne correspond pas à la valeur de hachage calculée sur le serveur, vous recevrez une erreur BadDigest. La réponse d'erreur inclut également des éléments de détail, la valeur de hachage que nous avions calculée et celle que vous nous aviez suggérée. Pendant le développement, vous pouvez utiliser ces informations pour diagnostiquer l'erreur. En production, un programme performant doit comprendre ces informations dans son journal des erreurs.

Réponse d'erreur SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Dans SOAP, un résultat d'erreur est renvoyé au client comme une erreur SOAP avec le code réponse HTTP 500. Si vous ne recevez pas une erreur SOAP, alors votre demande a abouti. Le

code d'erreur SOAP Amazon S3 se compose d'un code d'erreur SOAP 1.1 standard (« Server » ou « Client ») concaténé avec le code d'erreur spécifique Amazon S3. A titre d'exemple : « Server.InternalError » ou « Client.NoSuchBucket ». L'élément de chaîne d'erreur SOAP contient un message d'erreur générique directement lisible en anglais. Finalement, l'élément détaillé de l'erreur SOAP contient différentes informations concernant l'erreur.

A titre d'exemple, si vous tentez de supprimer l'objet « Fred », qui n'existe pas, le corps de la réponse SOAP contient une erreur SOAP « NoSuchKey ».

Example

```
<soapenv:Body>
  <soapenv:Fault>
    <Faultcode>soapenv:Client.NoSuchKey</Faultcode>
    <Faultstring>The specified key does not exist.</Faultstring>
    <Detail>
      <Key>Fred</Key>
    </Detail>
  </soapenv:Fault>
</soapenv:Body>
```

Pour de plus amples informations sur les erreurs Amazon S3, veuillez consulter ErrorCodeList.

Bonnes pratiques concernant les erreurs Amazon S3

Lors de la conception d'une application devant être utilisée avec Amazon S3, il est important de gérer les erreurs Amazon S3 de façon appropriée. Cette section décrit les erreurs à prendre en compte lors de la conception de votre application.

Relancer en cas d'erreur InternalErrors

Les erreurs internes sont des erreurs qui se produisent au sein de l'environnement Amazon S3.

Les demandes qui reçoivent une réponse InternalError n'ont probablement pas été traitées. A titre d'exemple, si une demande PUT renvoie une erreur de type InternalError, une commande GET ultérieure peut récupérer l'ancienne valeur ou la valeur mise à jour.

Si Amazon S3 renvoie une réponse InternalError, relancez la demande.

Optimiser l'application pour les erreurs répétées SlowDown

Comme avec tout système distribué, S3 a des mécanismes de protection qui détectent une surconsommation involontaire des ressources et réagissent en conséquence. Les erreurs SlowDown peuvent se produire quand un taux de demande élevé déclenche l'un de ces mécanismes. Réduire le taux de demande diminuera ou éliminera des erreurs de ce type. De manière générale, la plupart des utilisateurs ne rencontrent pas fréquemment ce type d'erreur. Néanmoins, si vous voulez plus d'informations ou si vous rencontrez des erreurs SlowDown importantes ou inattendues, signalez-les sur le forum des développeurs Amazon S3 ou inscrivez-vous au AWS Support à l'adresse https://aws.amazon.com/premiumsupport/.

Isoler les erreurs

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Amazon S3 fournit un ensemble de codes d'erreur qui sont utilisés par les deux API SOAP et REST. L'API SOAP renvoie des codes d'erreur Amazon S3 standard. L'API REST est conçue pour fonctionner comme un serveur HTTP standard et interagir avec les clients HTTP existants (par exemple, les navigateurs, les bibliothèques client HTTP, les serveurs proxy, les mémoires cache, etc.). Pour s'assurer que les clients HTTP gèrent correctement les erreurs, nous lions chaque erreur Amazon S3 à un code de statut HTTP.

Les codes de statut HTTP sont moins parlants que les codes d'erreur Amazon S3 et contiennent moins d'informations sur l'erreur. Par exemple, les erreurs Amazon S3 `NoSuchKey` et `NoSuchBucket` sont toutes les deux liées au code de statut `HTTP 404 Not Found`.

Même si les codes de statut HTTP contiennent moins d'informations sur l'erreur, les clients qui comprennent les codes HTTP, mais pas l'API Amazon S3, sont généralement à même de gérer l'erreur correctement.

Par conséquent, lors de la gestion d'erreurs ou du signalement des erreurs Amazon S3 aux utilisateurs finaux, utilisez le code d'erreur Amazon S3 à la place du code de statut HTTP, car il

contient plus d'informations sur l'erreur. De même, lors du débogage de votre application, vous devrez également consulter l'élément lisible <Details> de la réponse d'erreur XML.

Référence pour développeurs

Cette annexe inclut les sections suivantes.

Rubriques

Annexe A : Utilisation de l'API SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Au lieu d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Cette section contient des informations spécifiques à l'API SOAP Amazon S3.

> **Note**
>
> Les demandes SOAP, aussi bien authentifiées qu'anonymes, doivent être envoyées à Amazon S3 à l'aide de SSL. Amazon S3 renvoie un message d'erreur lorsque vous envoyez une demande SOAP via HTTP.

Rubriques

Éléments communs de l'API SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Vous pouvez interagir avec Amazon S3 en utilisant SOAP 1.1 sur HTTP. Le WSDL Amazon S3, qui décrit l'API Amazon S3 d'une manière lisible par machine, est disponible à l'adresse https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl. Le schéma Amazon S3 est disponible à l'adresse https://doc.s3.amazonaws.com/2006-03-01/AmazonS3.xsd.

La plupart des utilisateurs interagissent avec Amazon S3 à l'aide de la boîte à outils SOAP adaptée à leur langage et à leur environnement de développement. L'API Amazon S3 se présente de différentes façons selon les boîtes à outils. Veuillez vous reporter à la documentation propre à votre boîte à outils pour savoir comment l'utiliser. Cette section décrit les opérations SOAP d'Amazon S3 de façon neutre par rapport à la boîte à outils, en présentant les demandes et les réponses XML telles qu'elles apparaissent sur le réseau.

Éléments communs

Vous pouvez inclure les éléments suivants relatifs à l'autorisation dans une demande SOAP :

- `AWSAccessKeyId:` ID de clé d'accès AWS du demandeur
- `Timestamp:` heure actuelle sur votre système
- `Signature:` signature de la demande

Authentification des demandes SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Plutôt que d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les kits SDK AWS.

Chaque demande non anonyme doit contenir des informations d'authentification pour établir l'identité de la personne habilitée créant la demande. Dans SOAP, les informations d'authentification sont placées dans les éléments suivants de la demande SOAP :

- Votre ID de clé d'accès AWS

> **Note**
>
> Lorsque des demandes SOAP authentifiées sont créées, les informations d'identification temporaires ne sont pas prises en charge. Pour obtenir plus d'informations sur les types d'informations d'identification, consultez Demandes.

- `Timestamp:` Ceci doit être une dateTime (accédez à http://www.w3.org/TR/xmlschema-2/#dateTime) dans le fuseau horaire Heure universelle coordonnée (Heure de Greenwich), comme `2009-01-01T12:00:00.000Z`. L'autorisation échouera si cet horodatage a plus de 15 minutes d'écart avec l'horloge sur les serveurs Amazon S3.
- `Signature:` Hachage RFC 2104 HMAC-SHA1 (consultez http://www.ietf.org/rfc/rfc2104.txt) de la concaténation de « AmazonS3 » + OPERATION + horodatage, utilisant votre clé d'accès secrète AWS comme clé. Par exemple, dans l'exemple de demande CreateBucket, l'élément de signature contiendrait le hachage HMAC-SHA1 de la valeur « AmazonS3CreateBucket2009-01-01T12:00:00.000Z » :

Par exemple, dans l'exemple de demande CreateBucket, l'élément de signature contiendrait le hachage HMAC-SHA1 de la valeur « AmazonS3CreateBucket2009-01-01T12:00:00.000Z » :

Example

```
<CreateBucket xmlns="https://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Acl>private</Acl>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2009-01-01T12:00:00.000Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</CreateBucket>
```

> **Note**
>
> Les demandes SOAP, aussi bien authentifiées qu'anonymes, doivent être envoyées à Amazon S3 à l'aide de SSL. Amazon S3 renvoie un message d'erreur lorsque vous envoyez une demande SOAP via HTTP.

> **Important**
>
> En raison des diverses propositions existant sur la façon de renoncer à une précision horaire supplémentaire, les utilisateurs de .NET doivent faire attention à ne pas envoyer des horodatages Amazon S3 trop spécifiques. Pour y parvenir, il est possible de créer manuellement des objets `DateTime` avec une précision limitée aux millisecondes.

Configuration d'une stratégie d'accès avec SOAP

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Au lieu d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Le contrôle d'accès peut être défini au moment de l'écriture d'un bucket ou d'un objet en incluant l'élément AccessControlList « » dans la demande à `CreateBucketPutObjectInline`, ou`PutObject`. L' AccessControlList élément est décrit dansIdentity and Access Management pour Amazon S3. Si aucune liste de contrôle d'accès n'est spécifiée lors de ces opérations, la ressource est créée avec une politique d'accès par défaut qui donne au demandeur un accès FULL_CONTROL (c'est le cas même s'il s'agit d'une demande PutObjectInline ou d'une PutObject demande pour un objet qui existe déjà).

Vous trouverez ci-dessous une demande qui écrit des données dans un objet, rend l'objet lisible par des mandataires anonymes et donne à l'utilisateur spécifié les droits FULL_CONTROL sur le compartiment (la plupart des développeurs voudront s'accorder un accès FULL_CONTROL à leur propre compartiment).

Example

Vous trouverez ci-dessous une demande qui écrit des données dans un objet et rend l'objet lisible par des mandataires anonymes.

Sample Request

```
<PutObjectInline xmlns="https://doc.s3.amazonaws.com/2006-03-01">
  <Bucket>quotes</Bucket>
  <Key>Nelson</Key>
  <Metadata>
    <Name>Content-Type</Name>
    <Value>text/plain</Value>
  </Metadata>
  <Data>aGEtaGE=</Data>
  <ContentLength>5</ContentLength>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
        <DisplayName>chriscustomer</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers<URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
  </AccessControlList>
  <AWSAccessKeyId>AKIAIOSFODNN7EXAMPLE</AWSAccessKeyId>
  <Timestamp>2009-03-01T12:00:00.183Z</Timestamp>
  <Signature>Iuyz3d3P0aTou39dzbqaEXAMPLE=</Signature>
</PutObjectInline>
```

Sample Response

```
<PutObjectInlineResponse xmlns="https://s3.amazonaws.com/doc/2006-03-01">
  <PutObjectInlineResponse>
    <ETag>&quot828ef3fdfa96f00ad9f27c383fc9ac7f&quot</ETag>
    <LastModified>2009-01-01T12:00:00.000Z</LastModified>
  </PutObjectInlineResponse>
```

```
</PutObjectInlineResponse>
```

La stratégie de contrôle d'accès peut être lue ou spécifiée pour un compartiment ou un objet existant à l'aide des méthodes `GetBucketAccessControlPolicy`, `GetObjectAccessControlPolicy`, `SetBucketAccessControlPolicy` et `SetObjectAccessControlPolicy`. Pour de plus amples informations, veuillez consulter la description détaillée de ces méthodes.

Annexe b : Authentification des demandes (AWS signature version 2)

Important

Cette section décrit comment authentifier les demandes à l'aide de AWS Signature Version 2. Signature Version 2 est en cours de désactivation (obsolète), Amazon S3 acceptera uniquement les requêtes d'API signées avec Signature Version 4. Pour de plus amples informations, veuillez consulter AWS Signature version 2 désactivée (obsolète) pour Amazon S3

La version 4 de Signature est prise en charge dans toutes les régions Régions AWS, et c'est la seule version compatible avec les nouvelles régions. Pour plus d'informations, consultez Authentification des demandes (AWS Signature version 4) dans le manuel Amazon Simple Storage Service API Reference.

Amazon S3 vous permet d'identifier la version de signature d'API qui a été utilisée pour signer une demande. Il est important de déterminer si l'un de vos workflows utilise Signature Version 2 et de procéder à une mise à niveau vers Signature Version 4 afin d'éviter tout impact sur votre entreprise.

- Si vous utilisez des journaux d' CloudTrail événements (option recommandée), veuillez voir Identification des demandes Amazon S3 Signature version 2 à l'aide de CloudTrail comment interroger et identifier de telles demandes.
- Si vous utilisez les journaux Amazon S3 Server Access, veuillez consulter Identification des demandes Signature Version 2 à l'aide des journaux d'accès Amazon S3

Rubriques

Authentification des demandes grâce à l'API REST

Lorsque vous accédez à Amazon S3 grâce à REST, vous devez fournir les éléments suivants dans la demande afin qu'elle puisse être authentifiée :

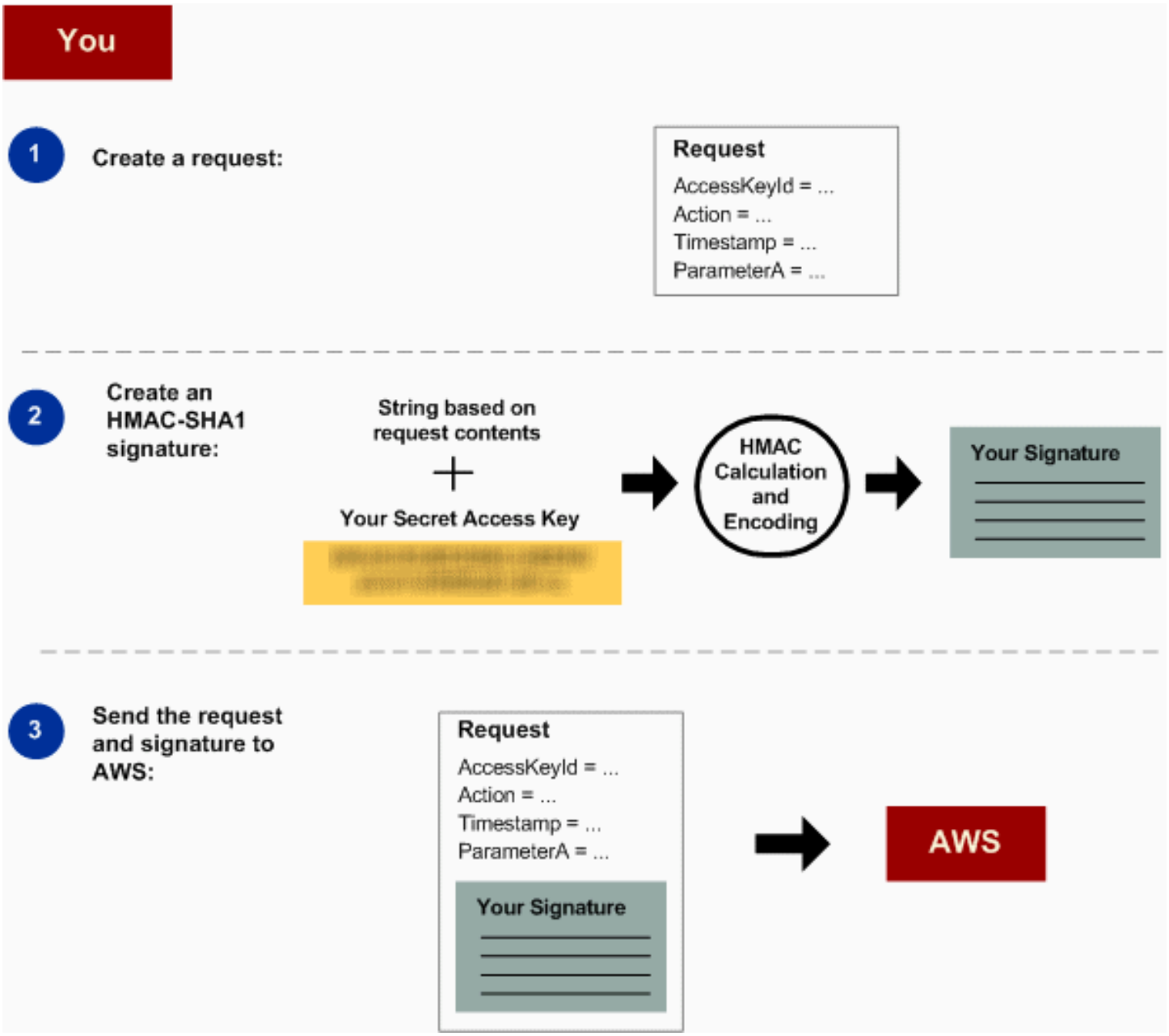
Éléments d'une demande

- AWS ID de clé d'accès — Chaque demande doit contenir l'ID de clé d'accès de l'identité que vous utilisez pour envoyer votre demande.
- Signature – Chaque demande doit contenir une signature de demande valide, sinon, la demande est rejetée.

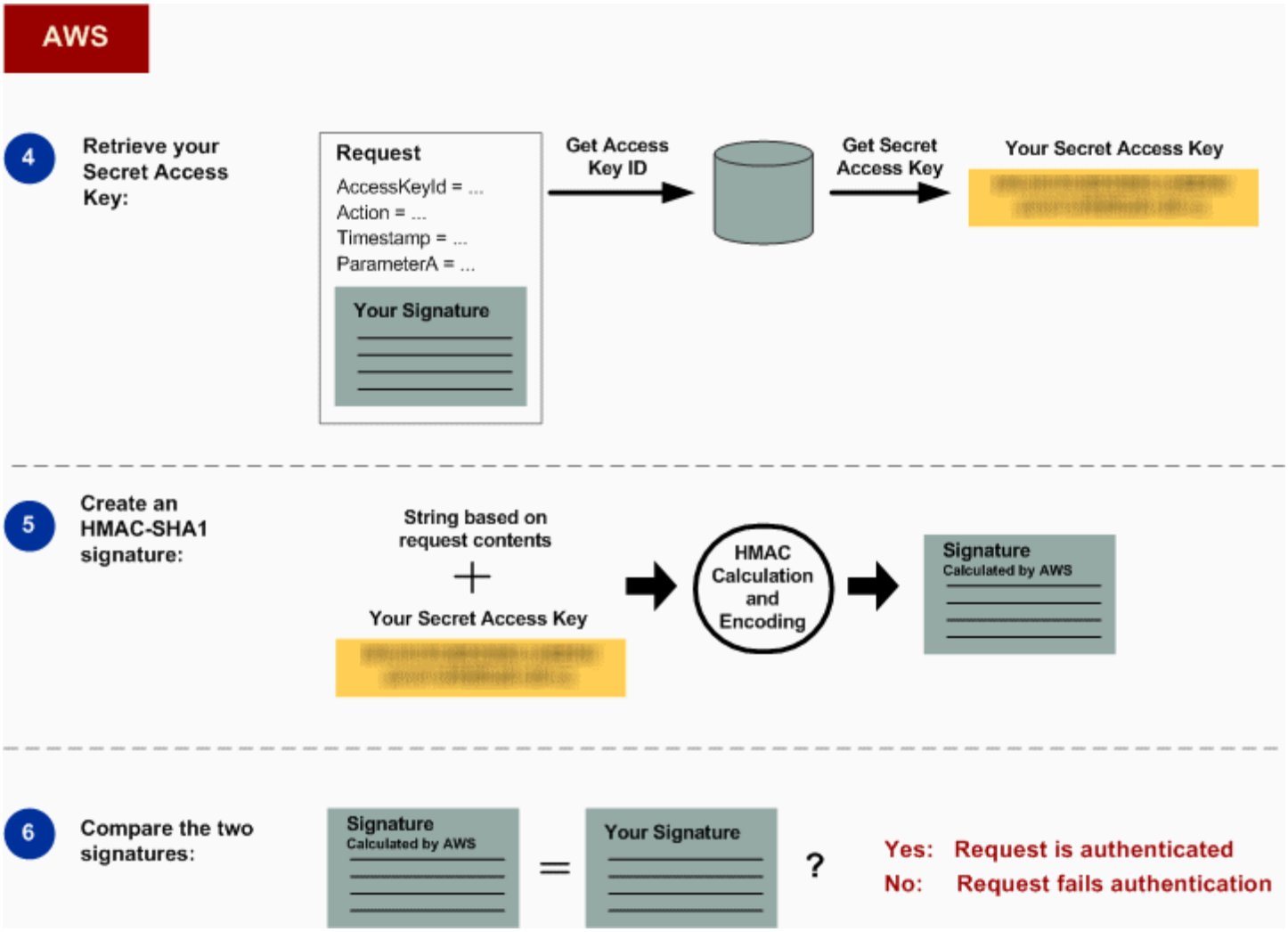
  Une signature de demande est définie grâce à la clé d'accès secrète, clé qui est connue uniquement par vous et AWS.
- Horodatage – Chaque demande doit contenir la date et l'heure de création de la demande sous la forme d'une chaîne de caractères en UTC.
- Date – Toute demande doit contenir son horodatage.

  Selon l'action de l'API que vous utilisez, vous pouvez fournir une date et une heure d'expiration de la demande au lieu ou en plus de l'horodatage. Consultez la rubrique sur l'authentification concernant l'action particulière pour déterminer ce qu'elle nécessite.

Voici les étapes générales pour authentifier des demandes sur Amazon S3. Nous supposons que vous disposez des informations d'identification de sécurité nécessaires, de l'ID de clé d'accès et de la clé d'accès secrète.



1	Créez une demande pour AWS.
2	Calculez la signature grâce à la clé d'accès secrète.
3	Envoyez la demande à Amazon S3. Incluez l'ID de clé d'accès et la signature dans la demande. Amazon S3 réalise les trois prochaines étapes.



4	Amazon S3 utilise l'ID de clé d'accès pour rechercher la clé d'accès secrète.
5	Amazon S3 calcule une signature à partir des données de la demande et de la clé d'accès secrète grâce au même algorithme que vous avez utilisé pour calculer la signature envoyée dans la demande.
6	Si la signature générée par Amazon S3 correspond à celle figurant dans la demande que vous avez envoyée, la demande est considérée comme authentique. Si la comparaison échoue, la demande est rejetée, et Amazon S3 renvoie une réponse d'erreur.

Informations d'authentification détaillées

Pour obtenir des informations détaillées sur l'authentification REST, consultez Signature et authentification des demandes REST.

Signature et authentification des demandes REST

Rubriques

> **Note**
>
> Cette rubrique explique les demandes d'authentification à l'aide de la Signature Version 2. Amazon S3 prend désormais en charge le dernier processus Signature Version 4. Ce protocole récent est pris en charge dans toutes les Régions et chaque nouvelle Région après le 30 janvier 2014 prendra uniquement en charge Signature Version 4. Pour de plus amples informations, veuillez consulter Demandes d'authentification (AWS Signature Version 4) dans la Référence API Amazon Simple Storage Service.

L'authentification est le processus qui consiste à prouver l'identité au système. L'identité est un facteur important dans les décisions liées au contrôle d'accès Amazon S3. Les demandes sont autorisées ou refusées en partie en fonction l'identité du demandeur. Par exemple, le droit de créer des compartiments est réservé aux développeurs inscrits et (par défaut) le droit de créer des objets dans un compartiment est réservé au propriétaire du compartiment en question. En tant que développeur, vous faites des demandes qui appellent ces privilèges, vous devez donc prouver

l'identité au système grâce à l'authentification des demandes. Cette section vous montre comment le faire.

> **Note**
>
> Le contenu de cette section ne s'applique pas à HTTP POST. Pour de plus amples informations, veuillez consulter Importations basées sur un navigateur à l'aide de POST (version de AWS signature 2).

L'API REST Amazon S3 utilise un schéma HTTP personnalisé basé sur un HMAC (Hash Message Authentication Code) à clés pour l'authentification. Pour authentifier une demande, vous devez d'abord concaténer les éléments sélectionnés de la demande pour forme une chaîne. Utilisez ensuite la clé d'accès secrète AWS pour calculer le HMAC de cette chaîne. Officieusement, nous appelons ce processus « signature de la demande », et nous appelons le résultat de l'algorithme HMAC la signature, car elle imite les propriétés de sécurité d'une vraie signature. Enfin, vous ajoutez cette signature comme un paramètre de la demande grâce à la syntaxe décrite dans cette section.

Lorsque le système reçoit une demande authentifiée, il récupère la clé d'accès secrète AWS que vous dites posséder pour l'utiliser de la même manière afin de calculer une signature pour le message qu'il reçoit. Il compare ensuite la signature calculée et la signature présentée par le demandeur. Si les deux signatures correspondent, le système conclut que le demandeur doit avoir accès à la clé d'accès secrète AWS, et agit donc avec l'autorité du mandataire pour qui la clé a été émise. Si les deux signatures ne correspondent pas, la demande est abandonnée et le système répond par un message d'erreur.

Example Demande REST Amazon S3 authentifiée

```
GET /photos/puppy.jpg HTTP/1.1
Host: awsexamplebucket1.us-west-1.s3.amazonaws.com
Date: Tue, 27 Mar 2007 19:36:42 +0000

Authorization: AWS AKIAIOSFODNN7EXAMPLE:
qgk2+6Sv9/oM7G3qLEjTH1a1l1g=
```

Utilisation d'informations d'identification de sécurité temporaires

Si vous signez la demande grâce à des informations d'identification de sécurité temporaires (consultez Demandes), vous devez inclure le token de sécurité correspondant dans la demande et ajouter l'en-tête `x-amz-security-token`.

Lorsque vous obtenez des informations d'identification de sécurité temporaires grâce à l'API AWS Security Token Service, la réponse inclut des informations d'identification de sécurité temporaires et un token de session. Vous indiquez la valeur du jeton de session dans l'en-tête `x-amz-security-token` lorsque vous envoyez des demandes à Amazon S3. Pour de plus amples informations sur l'API AWS Security Token Service fournie par IAM, veuillez consulter Action dans le Guide de référence API AWS Security Token Service.

L'en-tête Authentification

L'API REST Amazon S3 utilise l'en-tête `Authorization` HTTP standard pour transmettre les informations d'authentification. (Le nom de l'en-tête standard est maladroit car il comporte des informations d'authentification, pas d'autorisation.) Dans le schéma d'authentification Amazon S3, l'en-tête Autorisation possède la forme suivante :

```
Authorization: AWS AWSAccessKeyId:Signature
```

Les développeurs reçoivent un ID de clé d'accès AWS et une clé d'accès secrète AWS quand ils s'inscrivent. Pour une authentification de demande, l'élément `AWSAccessKeyId` identifie l'ID de clé d'accès utilisé pour calculer la signature et, indirectement, le développeur à l'origine de la demande.

L'élément `Signature` est le RFC 2104 HMAC-SHA1 des éléments sélectionnés issus de la demande, et donc la partie `Signature` de l'en-tête Autorisation varie d'une demande à l'autre. Si la signature de la demande calculée par le système correspond à la `Signature` incluse dans la demande, cela prouve que le demandeur possède la clé d'accès secrète AWS. La demande sera ensuite traitée sous l'identité, et avec l'autorité, du développeur pour qui la clé a été émise.

Voici une pseudo-grammaire qui illustre la construction de l'en-tête de la demande `Authorization`. (Dans l'exemple, \n signifie le point de code Unicode `U+000A`, communément appelé nouvelle ligne).

```
Authorization = "AWS" + " " + AWSAccessKeyId + ":" + Signature;

Signature = Base64( HMAC-SHA1( UTF-8-Encoding-Of(YourSecretAccessKey), UTF-8-Encoding-
Of( StringToSign ) ) );

StringToSign = HTTP-Verb + "\n" +
 Content-MD5 + "\n" +
 Content-Type + "\n" +
 Date + "\n" +
 CanonicalizedAmzHeaders +
 CanonicalizedResource;
```

```
CanonicalizedResource = [ "/" + Bucket ] +
 <HTTP-Request-URI, from the protocol name up to the query string> +
 [ subresource, if present. For example "?acl", "?location", or "?logging"];

CanonicalizedAmzHeaders = <described below>
```

HMAC-SHA1 est un algorithme défini par RFC 2104 - le hachage à clés pour l'authentification d'un message. L'algorithme utilise comme paramètres des chaînes encodées sur deux octets, une clé et un message. Pour une authentification de la demande Amazon S3, utilisez votre clé d'accès secrète AWS (`YourSecretAccessKey`) comme clé, et l'encodage UTF-8 de `StringToSign` comme message. Le résultat de HMAC-SHA1 est également une chaîne d'octets, appelée valeur de hachage. Le paramètre de la demande `Signature` est construit en encodant en Base64 cette valeur de hachage.

Conversion sous forme canonique d'une demande pour signature

Rappelez-vous que lorsque le système reçoit une demande authentifiée, il compare la signature calculée de la demande et celle fournie dans la demande dans `StringToSign`. C'est pourquoi, vous devez calculer la signature grâce à la même méthode utilisée par Amazon S3. Le processus qui consiste à mettre une demande dans une forme définie pour la signature est appelé conversion sous forme canonique.

Construction de l' CanonicalizedResource élément

`CanonicalizedResource` représente la ressource Amazon S3 ciblée par la demande. Construisez-la pour une demande REST de la manière suivante :

Lancer le traitement

1	Commencez par une chaîne de caractères vide ("").
2	Si la demande spécifie un compartiment qui utilise l'en-tête hôte HTTP (demande d'héberge ment virtuel), ajoutez le nom du compartiment précédé d'un "/" (par exemple, "/nom du compartiment"). Pour les demandes de type chemin et les demandes qui ne font pas référence à un compartiment, ne faites rien. Pour en savoir plus sur les demandes d'héberge ment virtuel, consultez Hébergement virtuel de compartiments. Pour une demande de type hébergement virtuel « https://awsexamplebucket1.s3.us-west -1.amazonaws.com/photos/puppy.jpg », le `CanonicalizedResource` est « /awsexamp lebucket1 ».

	Pour une demande de type chemin d'accès, « https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg », le `CanonicalizedResource` est « ».
3	Ajoutez le chemin de la demande-URI HTTP non décodée, jusqu'à la chaîne d'interrogation mais sans l'inclure. Pour une demande de type hébergement virtuel « https://awsexamplebucket1.s3.us-west-1.amazonaws.com/photos/puppy.jpg », le `CanonicalizedResource` est « /awsexamplebucket1/photos/puppy.jpg ». Pour une demande de style chemin, « https://s3.us-west-1.amazonaws.com/awsexamplebucket1/photos/puppy.jpg », l'élément `CanonicalizedResource` est « /awsexamplebucket1/photos/puppy.jpg ». A ce stade, la `CanonicalizedResource` est identique pour la demande d'hébergement virtuel et de type chemin. Pour une demande qui ne fait pas référence à un compartiment, comme GET Service, ajoutez « / ».
4	Si la demande fait référence à une sous-ressource, comme `?versioning` , `?location` , `?acl`, `?lifecycle` , `?versionid` , ou, ajoutez la sous-ressource, sa valeur le cas échéant et le point d'interrogation. Notez qu'en présence de plusieurs sous-ressources, les sous-ressources doivent être triées de manière lexicographique par nom de sous-ressource et séparées par « & », par exemple, ?acl&versionId=*valeur*. Les sous-ressources qui doivent être incluses lors de la construction de l' CanonicalizedResource élément sont acl, lifecycle, location, logging, notification, partNumber, policy, requestPayment, UploadID, uploads, versionId, versionId, versioning, versions et site web. Si la demande spécifie des paramètres de la chaîne d'interrogation qui ignorent les valeurs de l'en-tête de la réponse (veuillez consulter GetObject), ajoutez les paramètres de la chaîne d'interrogation et leurs valeurs. Lors de la signature, vous n'encodez pas ces valeurs ; toutefois, lorsque vous faites la demande, vous devez encoder les valeurs de ces paramètres. Les paramètres de la chaîne d'interrogation dans une demande GET incluent `response-content-type` , `response-content-language` , `response-expires` , `response-cache-control` , `response-content-disposition` et `response-content-encoding` . Le paramètre de chaîne de `delete` requête doit être inclus lorsque vous créez la demande de suppression CanonicalizedResource pour plusieurs objets.

Les éléments provenant de l' CanonicalizedResource URI de demande HTTP doivent être signés littéralement tels qu'ils apparaissent dans la requête HTTP, y compris les méta-caractères codés par URL.

La `CanonicalizedResource` doit être différente de la demande-URI HTTP. En particulier, si la demande utilise l'en-tête `Host` HTTP pour spécifier un compartiment, ce dernier n'apparaît pas dans la demande-URI HTTP. Toutefois, la `CanonicalizedResource` continue d'inclure le compartiment. Les paramètres de la chaîne d'interrogation doivent également apparaître dans la demande-URI mais ne sont pas inclus dans la `CanonicalizedResource`. Pour plus d'informations, consultez Hébergement virtuel de compartiments.

Construction de l' CanonicalizedAmzHeaders élément

Pour créer la CanonicalizedAmzHeaders partie de`StringToSign`, sélectionnez tous les en-têtes de requête HTTP commençant par « x-amz- » (en utilisant une comparaison qui ne distingue pas les majuscules et minuscules), puis appliquez le processus suivant.

CanonicalizedAmzHeaders processus

1	Convertissez chaque nom d'en-tête HTTP en minuscule. Par exemple, « `X-Amz-Date` » devient « `x-amz-date` ».
2	Triez la sélection d'en-tête de manière lexicographique par nom d'en-tête.
3	Combinez les champs d'en-tête portant le même nom dans une paire « header-name : comma-separated-value-list » comme prescrit par la RFC 2616, section 4.2, sans espaces entre les valeurs. Par exemple, les deux en-têtes de métadonnées « `x-amz-meta-username: fred` » et « `x-amz-meta-username: barney` » seraient combinés en un seul en-tête « `x-amz-meta-username: fred,barney` ».
4	« Séparez » les longs en-têtes qui s'étendent sur plusieurs lignes (comme le permet RFC 2616, section 4.2) en remplaçant le retour à la ligne (notamment la nouvelle ligne) par un espace simple.
5	Coupez les espaces autour des deux-points dans l'en-tête. Par exemple, l'en-tête « `x-amz-meta-username: fred,barney` » devient « `x-amz-meta-usernam e:fred,barney` »

6	Enfin, ajoutez un caractère de nouvelle ligne (U+000A) à chaque en-tête converti sous forme canonique dans la liste obtenue. Construisez l' CanonicalizedResource élément en concaténant tous les en-têtes de cette liste en une seule chaîne.

Éléments d'en-tête HTTP positionnels et éléments d'en-tête StringToSign HTTP nommés

Les tout premiers éléments de l'en-tête de `StringToSign` (Content-Type, Date et Content-MD5) sont positionnels par nature. `StringToSign` n'inclut pas les noms de ces en-têtes, uniquement leurs valeurs de la demande. En revanche, les éléments « `x-amz-` » sont nommés. Les noms et les valeurs de l'en-tête apparaissent dans `StringToSign`.

Si un en-tête positionnel demandé pour une définition de `StringToSign` n'est pas présent dans la demande (par exemple, les en-têtes `Content-Type` ou `Content-MD5` sont facultatifs pour les demandes PUT et insignifiants pour les demandes GET), substituez la chaîne de caractères vide ("") pour cette position.

Exigence d'horodatage

Un horodatage valide (qui utilise l'en-tête `Date` HTTP ou une alternative `x-amz-date`) est obligatoire pour les demandes authentifiées. De plus, l'horodatage du client inclus dans une demande authentifiée doit se faire dans les 15 minutes de l'heure du système Amazon S3 à la réception de la demande. Dans le cas contraire, la demande échoue avec le code d'erreur `RequestTimeTooSkewed`. L'objectif de ces restrictions est de limiter la possibilité que des demandes interceptées soient réutilisées par un adversaire. Pour une meilleure protection contre l'écoute, utilisez le transport HTTPS pour des demandes authentifiées.

> **Note**
>
> La contrainte de validation sur la date de la demande s'applique uniquement aux demandes authentifiées qui n'utilisent pas d'authentification par chaîne d'interrogation. Pour de plus amples informations, veuillez consulter Alternative à l'authentification d'une demande par chaîne d'interrogation.

Certaines bibliothèques client HTTP n'offrent pas la possibilité de configurer l'en-tête `Date` pour une demande. Si vous avez des problèmes pour inclure la valeur de l'en-tête « Date » dans les en-têtes convertis sous forme canonique, vous pouvez à la place configurer l'horodatage pour la demande grâce à un en-tête « `x-amz-date` ». La valeur de l'en-tête `x-amz-date` doit être dans l'un des

formats RFC 2616 (http://www.ietf.org/rfc/rfc2616.txt). Lorsqu'un en-tête `x-amz-date` est présent dans une demande, le système ignore tous les en-têtes `Date` lors du calcul de la signature de la demande. Par conséquent, si vous incluez l'en-tête `x-amz-date`, utilisez la chaîne de caractères vide pour la `Date` lors de la création de l'élément `StringToSign`. consultez suivante pour obtenir un exemple.

Exemples d'authentification

Les exemples de cette section utilisent les informations d'identification (qui ne fonctionnent pas) dans le tableau suivant.

Paramètre	Valeur
AWSAccessKeyId	`AKIAIOSFODNN7EXAMPLE`
AWSSecret AccessKey	`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

Dans l'exemple `StringToSign`, le formatage n'est pas significatif, et `\n` signifie le point de code Unicode `U+000A`, communément appelé nouvelle ligne. De plus, les exemples utilisent « +0000 » pour désigner le fuseau horaire. Vous pouvez utiliser « GMT » à la place pour désigner le fuseau horaire, mais les signatures illustrées dans les exemples sont différentes.

Objet GET

Cet exemple obtient un objet à partir du compartiment awsexamplebucket1.

Demande	StringToSign
`GET /photos/puppy.jpg HTTP/1.1` `Host: awsexamplebucket1.us-west-1.s3.amazonaws.com` `Date: Tue, 27 Mar 2007 19:36:42 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE: qgk2+6Sv9/oM7G3qLEjTH1a1l1g=`*	`GET\n` `\n` `\n` `Tue, 27 Mar 2007 19:36:42 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Notez que le nom du compartiment est CanonicalizedResource inclus, mais pas l'URI de demande HTTP. (Le compartiment est spécifié par l'en-tête hôte.)

> **Note**
>
> Le script Python suivant calcule la signature précédente à l'aide des paramètres fournis. Vous pouvez utiliser ce script pour créer vos propres signatures, en remplaçant les clés, le cas StringToSign échéant.

```
import base64
import hmac
from hashlib import sha1

access_key = 'AKIAIOSFODNN7EXAMPLE'.encode("UTF-8")
secret_key = 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY'.encode("UTF-8")

string_to_sign = 'GET\n\n\nTue, 27 Mar 2007 19:36:42 +0000\n/awsexamplebucket1/
photos/puppy.jpg'.encode("UTF-8")
signature = base64.b64encode(
                                hmac.new(
                                        secret_key, string_to_sign, sha1
                                        ).digest()
                                ).strip()


print(f"AWS {access_key.decode()}:{signature.decode()}")
```

Objet PUT

Cet exemple place un objet dans le compartiment awsexamplebucket1.

Demande	StringToSign
`PUT /photos/puppy.jpg HTTP/1.1` `Content-Type: image/jpeg` `Content-Length: 94328` `Host: awsexamplebucket1.s3.us-wes` `t-1.amazonaws.com` `Date: Tue, 27 Mar 2007 21:15:45 +0000`	`PUT\n` `\n` `image/jpeg\n` `Tue, 27 Mar 2007 21:15:45 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Demande	StringToSign
Authorization: AWS AKIAIOSFODNN7EXAMP LE: iqRzw+ileNPu1fhspnRs8nOjjIA=	

Notez l'en-tête Content-Type dans la demande et dans le. StringToSign Notez également que le champ Content-MD5 est laissé vide dans le StringToSign, car il n'est pas présent dans la demande.

Liste

Cet exemple répertorie le contenu du compartiment awsexamplebucket1.

Demande	StringToSign
GET /?prefix=photos&max-keys=50&marker=puppy HTTP/1.1 User-Agent: Mozilla/5.0 Host: awsexamplebucket1.s3.us-west-1.amazonaws.com Date: Tue, 27 Mar 2007 19:42:41 +0000 *Authorization: AWS AKIAIOSFODNN7EXAMPLE: m0WP8eCtspQl5Ahe6L1SozdX9YA=*	GET\n \n \n Tue, 27 Mar 2007 19:42:41 +0000\n /awsexamplebucket1/

Notez la barre oblique finale CanonicalizedResource et l'absence de paramètres de chaîne de requête.

Fetch

Cet exemple récupère la stratégie de contrôle d'accès pour le compartiment « awsexamplebucket1 ».

Demande	StringToSign
GET /?acl HTTP/1.1	GET\n \n

Demande	StringToSign
`Host: awsexamplebucket1.s3.us-west-1.amazonaws.com` `Date: Tue, 27 Mar 2007 19:44:46 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE:82ZHiFIjc+WbcwFKGUVEQspPn+0=`*	`\n` `Tue, 27 Mar 2007 19:44:46 +0000\n` `/awsexamplebucket1/?acl`

Remarquez comment le paramètre de chaîne de requête de sous-ressource est inclus dans le CanonicalizedResource.

Suppression

Cet exemple supprime un objet du compartiment « awsexamplebucket1 » grâce à l'alternative de type chemin et Date.

Demande	StringToSign
`DELETE /awsexamplebucket1/photos/puppy.jpg HTTP/1.1` `User-Agent: dotnet` `Host: s3.us-west-1.amazonaws.com` `Date: Tue, 27 Mar 2007 21:20:27 +0000` `x-amz-date: Tue, 27 Mar 2007 21:20:26 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE:XbyTlbQdu9Xw5o8P4iMwPktxQd8=`*	`DELETE\n` `\n` `\n` `Tue, 27 Mar 2007 21:20:26 +0000\n` `/awsexamplebucket1/photos/puppy.jpg`

Notez comment nous avons utilisé la méthode alternative x-amz-date « » pour spécifier la date (parce que notre bibliothèque cliente nous empêchait de définir la date, par exemple). Dans ce cas, le code `x-amz-date` prime sur l'en-tête `Date`. Par conséquent, l'entrée de la date dans la signature doit contenir la valeur de l'en-tête `x-amz-date`.

Charger

Cet exemple charge un objet dans un compartiment d'hébergement virtuel CNAME avec des métadonnées.

Demande	StringToSign
`PUT /db-backup.dat.gz HTTP/1.1` `User-Agent: curl/7.15.5` `Host: static.example.com:8080` `Date: Tue, 27 Mar 2007 21:06:08 +0000` `x-amz-acl: public-read` `content-type: application/x-download` `Content-MD5: 4gJE4saaMU4BqNR0kLY+lw==` `X-Amz-Meta-ReviewedBy: joe@example.com` `X-Amz-Meta-ReviewedBy: jane@exam` `ple.com` `X-Amz-Meta-FileChecksum: 0x02661779` `X-Amz-Meta-ChecksumAlgorithm: crc32` `Content-Disposition: attachment;` `filename=database.dat` `Content-Encoding: gzip` `Content-Length: 5913339` *`Authorization: AWS AKIAIOSFODNN7EXAMP`* *`LE:`* *`jtBQa0Aq+DkULFI8qrpwIjGEx0E=`*	`PUT\n` `4gJE4saaMU4BqNR0kLY+lw==\n` `application/x-download\n` `Tue, 27 Mar 2007 21:06:08 +0000\n` `x-amz-acl:public-read\n` `x-amz-meta-checksumalgorithm:c` `rc32\n` `x-amz-meta-filechecksum:0x026` `61779\n` `x-amz-meta-reviewedby:` `joe@example.com,jane@example.com` `\n` `/static.example.com/db-backup.dat` `.gz`

Notez la façon dont les en-têtes sont triés, débarrassés de leurs espaces et convertis en minuscules. Notez également que plusieurs en-têtes avec le même nom ont été réunis grâce aux virgules pour séparer les valeurs.

Notez comme seuls les en-têtes d'entité HTTP `Content-Type` et `Content-MD5` apparaissent dans l'élément `StringToSign`. Les autres en-têtes d'entité `Content-*` n'apparaissent pas.

De nouveau, notez que la `CanonicalizedResource` inclut le nom du compartiment, ce qui n'est pas le cas de la demande-URI HTTP. (Le compartiment est spécifié par l'en-tête hôte.)

Lister tous les compartiments

Demande	StringToSign
`GET / HTTP/1.1`	`GET\n`

Demande	StringToSign
`Host: s3.us-west-1.amazonaws.com` `Date: Wed, 28 Mar 2007 01:29:59 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMPLE:qGdzdE`* *`RIC03wnaRNKh6OqZehG9s=`*	`\n` `\n` `Wed, 28 Mar 2007 01:29:59` `+0000\n` `/`

Clés Unicode

Demande	StringToSign
`GET /dictionary/fran%C3%A7ais/pr` `%c3%a9f%c3%a8re HTTP/1.1` `Host: s3.us-west-1.amazonaws.com` `Date: Wed, 28 Mar 2007 01:49:49 +0000` *`Authorization: AWS AKIAIOSFODNN7EXAMP`* *`LE:DNEZGsoieTZ92F3bUfSPQcbGmlM=`*	`GET\n` `\n` `\n` `Wed, 28 Mar 2007 01:49:49 +0000\n` `/dictionary/fran%C3%A7ais/pr` `%c3%a9f%c3%a8re`

> **Note**
>
> Les éléments dans `StringToSign` dérivés de la demande-URI sont pris littéralement, notamment l'encodage-URL et la casse.

Problèmes de signature de la demande REST

En cas d'échec de l'authentification de la demande REST, le système répond à la demande avec un rapport d'erreur XML. Les informations contenues dans ce rapport d'erreur aident les développeurs à identifier le problème. L'élément `StringToSign` du rapport d'erreur `SignatureDoesNotMatch` vous indique notamment la conversion sous forme canonique de la demande utilisée par le système.

Certaines boîtes à outils insèrent en silence et au préalable des en-têtes comme l'en-tête `Content-Type` lors d'une opération PUT. Dans la plupart des cas, la valeur de l'en-tête inséré reste constante, ce qui vous permet d'identifier les en-têtes manquants grâce à des outils comme Ethereal ou tcpmon.

Alternative à l'authentification d'une demande par chaîne d'interrogation

Vous pouvez authentifier certains types de demandes en transférant les informations requises en tant que paramètres de la chaîne d'interrogation au lieu d'utiliser l'en-tête HTTP `Authorization`. Cette méthode est utile pour activer l'accès direct d'un navigateur tiers vers les données Amazon S3 privées sans rediriger la demande. L'idée est de créer une demande « pré-signée » et de l'encoder comme une URL récupérable par le navigateur d'un utilisateur final. De plus, vous pouvez limiter une demande pré-signée en spécifiant un délai d'expiration.

Pour de plus amples informations sur l'utilisation des paramètres de requête pour authentifier les demandes, veuillez consulter Authentification des demandes à l'aide des paramètres de requête (AWS Signature Version 4) dans la Référence API Amazon Simple Storage Service. Pour obtenir des exemples d'utilisation des kits SDK AWS pour générer des URL pré-signées, veuillez consulter Partage d'objets à l'aide d'URL présignées.

Création d'une signature

Voici un exemple de requête REST Amazon S3 authentifiée par la chaîne d'interrogation.

```
GET /photos/puppy.jpg
?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Expires=1141889120&Signature=vjbyPxybdZaNmGa
%2ByT272YEAiv4%3D HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
Date: Mon, 26 Mar 2007 19:37:58 +0000
```

La méthode d'authentification d'une demande par chaîne d'interrogation n'exige aucun en-tête HTTP spécial. A la place, les éléments d'authentification requis sont spécifiés en tant que paramètres de la chaîne d'interrogation :

Nom du paramètre de la chaîne d'interrogation	Exemple de valeur	Description
`AWSAccessKeyId`	`AKIAIOSFODNN7EXAMPLE`	Votre ID de clé d'accès AWS. Il spécifie la clé d'accès secrète AWS utilisée pour signer la demande et,

Nom du paramètre de la chaîne d'interrogation	Exemple de valeur	Description
		indirectement, l'identité du développeur qui effectue la demande.
`Expires`	`1141889120`	Le délai d'expiration de la signature, indiqué en nombre de secondes depuis l'époque (00:00:00 UTC le 1er janvier 1970). Passé ce délai (selon le serveur), toute demande reçue est rejetée.
`Signature`	`vjbyPxybdZaNmGa%2B yT272YEAiv4%3D`	Le codage URL du codage Base64 du HMAC-SHA1 de. StringToSign

La méthode d'authentification de la demande par chaîne d'interrogation diffère légèrement du paramètre de la demande `Signature` et de l'élément `StringToSign`. Voici une pseudo-grammaire qui illustre la construction de la méthode d'authentification de la demande par chaîne d'interrogation.

```
Signature = URL-Encode( Base64( HMAC-SHA1( YourSecretAccessKey, UTF-8-Encoding-
Of( StringToSign ) ) ) );

StringToSign = HTTP-VERB + "\n" +
    Content-MD5 + "\n" +
    Content-Type + "\n" +
    Expires + "\n" +
    CanonicalizedAmzHeaders +
    CanonicalizedResource;
```

`YourSecretAccessKey` est l'ID de la clé d'accès secrète AWS qu'Amazon vous attribue lorsque vous vous inscrivez pour être un développeur Amazon Web Service. Notez la façon dont la `Signature` est encodée comme URL pour pouvoir être insérée dans la chaîne d'interrogation. Notez également que dans l'élément `StringToSign`, l'élément HTTP positionnel `Date` a été remplacé par `Expires`. Les `CanonicalizedAmzHeaders` et la `CanonicalizedResource` sont identiques.

> **Note**
>
> Dans la méthode d'authentification par chaîne d'interrogation, vous n'utilisez pas le `Date` ou l'en-tête `x-amz-date request` lors du calcul de l'élément StringToSign.

Authentification d'une demande par chaîne d'interrogation

Demande	StringToSign
`GET /photos/puppy.jpg?AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&` `    Signature=NpgCjnDzrM%2BWFzoENXmpNDUsSn8%3D&` `    Expires=1175139620 HTTP/1.1` `Host: awsexamplebucket1.s3.us-west-1.amazonaws.com`	`GET\n` `\n` `\n` `1175139620\n` `/awsexamplebucket1/photos/puppy.jpg`

Nous supposons que lorsqu'un navigateur effectue une demande GET, il ne fournit pas d'en-tête Content-MD5 ou Content-Type, et ne configure aucun en-tête x-amz-, ces parties de l'élément `StringToSign` restent donc vides.

Utilisation de l'encodage en Base64

Les signatures de la demande HMAC doivent être encodées en Base64. L'encodage Base64 convertit la signature en une simple chaîne de caractères ASCII qui peut être attachée à la demande. Les caractères qui peuvent apparaître dans la chaîne de signature comme plus (+), barre oblique (/) et égal (=) doivent être encodés s'ils sont utilisés dans une URI. Par exemple, si le code d'authentification inclut un signe plus (+), utilisez %2B dans la demande. Pour une barre oblique, utilisez %2F et pour le signe égal, utilisez %3D.

Pour obtenir des exemples d'encodage en Base64, consultez les Amazon S3 Exemples d'authentification.

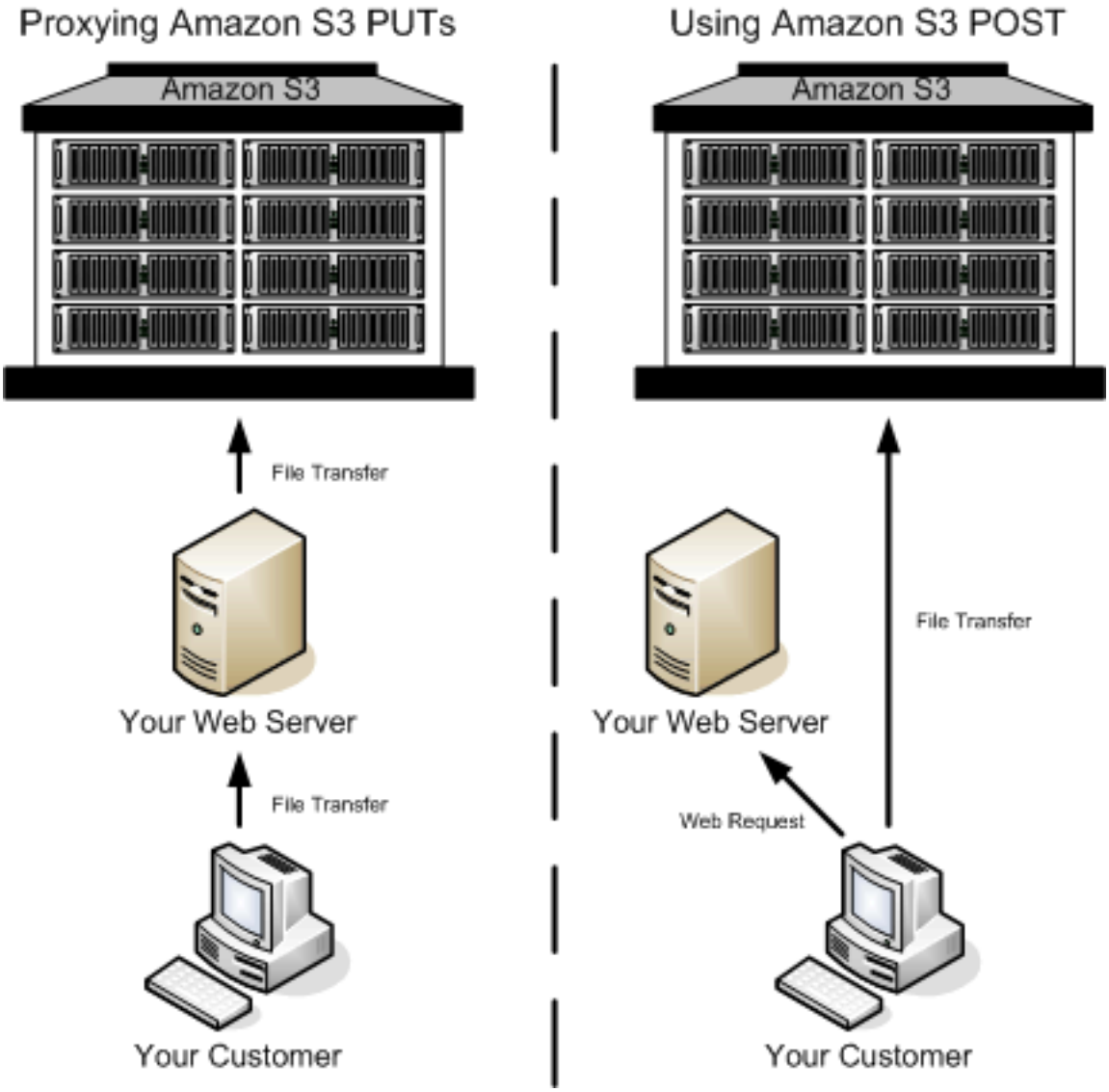
Importations basées sur un navigateur à l'aide de POST (version de AWS signature 2)

Amazon S3 prend en charge POST, qui permet à vos utilisateurs de télécharger directement du contenu vers Amazon S3. La méthode POST a été conçue pour simplifier les chargements, réduire la

latence de chargement et économiser de l'argent sur les applications dans lesquelles les utilisateurs chargent des données à stocker dans Amazon S3.

> **Note**
>
> L'authentification des demandes abordée dans cette section est basée sur AWS Signature Version 2, un protocole permettant d'authentifier les demandes d'API entrantes adressées aux AWS services.
> Amazon S3 prend désormais en charge la version 4 de Signature, un protocole permettant d'authentifier les demandes d'API entrantes adressées aux AWS services, dans l'ensemble. Régions AWSÀ l'heure actuelle, le protocole Régions AWS créé avant le 30 janvier 2014 continuera de prendre en charge le protocole précédent, Signature Version 2. Toutes les nouvelles Régions créées après le 30 janvier 2014 prendront en charge uniquement Signature Version 4. Par conséquent, toutes les demandes adressées à ces Régions doivent être effectuées avec Signature Version 4. Pour plus d'informations, consultez Authentification des demandes lors de téléchargements basés sur un navigateur à l'aide de POST (AWS Signature Version 4) dans le manuel Amazon Simple Storage Service API Reference.

La figure suivante illustre un téléchargement à l'aide de la méthode POST Amazon S3.



Chargement à l'aide de POST

1	L'utilisateur ouvre un navigateur web et accède à votre page web.
2	Votre page web comprend un formulaire HTTP qui contient toutes les informations nécessaires à l'utilisateur pour qu'il puisse télécharger le contenu sur Amazon S3.
3	L'utilisateur télécharge directement le contenu sur Amazon S3.

> **Note**
>
> L'authentification par chaîne d'interrogation n'est pas prise en charge pour POST.

Formulaires HTML (AWS signature version 2)

Rubriques

Lorsque vous communiquez avec Amazon S3, vous utilisez normalement l'API REST ou SOAP pour effectuer des opérations telles que put, get, delete, etc. Avec POST, les utilisateurs téléchargent les données directement sur Amazon S3 via leur navigateur, qui ne peut pas exécuter l'API SOAP ni créer de demande REST PUT.

> **Note**
>
> La prise en charge de SOAP sur HTTP est obsolète, mais SOAP continue d'être disponible sur HTTP. Les nouvelles fonctions Amazon S3 ne sont pas prises en charge pour SOAP. Au lieu d'utiliser SOAP, nous vous recommandons d'utiliser l'API REST ou les AWS SDK.

Utilisez les formulaires HTML pour autoriser les utilisateurs à télécharger du contenu vers Amazon S3 en utilisant leur navigateur. Les formulaires HTML se composent d'une déclaration de formulaire et de champs de formulaire. La déclaration de formulaire contient des informations de haut niveau sur la demande. Les champs de formulaire contiennent des informations détaillées sur la demande, ainsi que sur la stratégie utilisée pour authentifier cette dernière et garantir qu'elle répond aux conditions que vous spécifiez.

> **Note**
>
> Les limites et les données de formulaire (à l'exclusion du contenu du fichier) ne peuvent pas dépasser 20 Ko.

Cette section explique comment utiliser les formulaires HTML.

Encodage des formulaires HTML

Le formulaire et la stratégie doivent être encodés en UTF-8. Vous pouvez appliquer un encodage UTF-8 au formulaire en le spécifiant dans l'en-tête HTML ou comme en-tête de la demande.

> **Note**
>
> La déclaration d'un formulaire HTML n'accepte pas les paramètres d'authentification par chaîne d'interrogation.

Voici un exemple d'encodage en UTF-8 dans l'en-tête HTML :

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
```

Voici un exemple d'encodage en UTF-8 dans un en-tête de demande :

```
Content-Type: text/html; charset=UTF-8
```

Déclaration de formulaire HTML

La déclaration d'un formulaire possède trois composants : l'action, la méthode et le type d'encadrement. Si l'une quelconque de ces valeurs est définie de façon incorrecte, la demande échoue.

L'action spécifie l'URL qui doit traiter la demande et qu'il convient de définir sur l'URL du compartiment. Par exemple, si le nom de votre compartiment est `awsexamplebucket1` et que la Région est USA Ouest (Californie du Nord), l'URL est `https://awsexamplebucket1.s3.us-west-1.amazonaws.com/`.

> **Note**
>
> Le nom de clé est spécifié dans un champ du formulaire.

La méthode doit être POST.

Le type d'encadrement (enctype) doit être spécifié et défini sur multipart/form-data à la fois pour les chargements de fichiers et les chargements de zones de texte. Pour de plus amples informations, veuillez consulter la RFC 1867.

Example

L'exemple suivant est une déclaration de formulaire pour le compartiment « awsexamplebucket1 ».

```
<form action="https://awsexamplebucket1.s3.us-west-1.amazonaws.com/" method="post"

enctype="multipart/form-data">
```

Champs de formulaire HTML

Le tableau ci-dessous décrit les champs qui peuvent être utilisés au sein d'un formulaire HTML.

> **Note**
>
> La variable `${filename}` est automatiquement remplacée par le nom du fichier fourni par l'utilisateur et est reconnue par l'ensemble des champs de formulaire. Si le navigateur ou le client fournit un chemin d'accès complet ou partiel au fichier, seul le texte suivant la dernière barre oblique (/) ou barre oblique inverse (\) sera utilisé. Par exemple, « C:\Program Files \dossier1\fichier.txt » est interprété comme « fichier.txt ». Si aucun fichier ni nom de fichier n'est fourni, la variable est remplacée par une chaîne vide.

Nom de champ	Description	Obligatoire
`AWSAccessKeyId`	ID de clé d' AWS accès du propriétaire du compartiment qui accorde à un utilisateur anonyme l'accès à une demande répondant	Conditionnel

Nom de champ	Description	Obligatoire
	à l'ensemble des contraintes de la politique. Ce champ est requis si la demande inclut un document de stratégie.	
`acl`	Une liste de contrôle d'accès (ACL) Amazon S3 Si une liste de contrôle d'accès non valide est spécifiée, une erreur est générée. Pour plus d'informations sur les listes ACL, consultez Listes de contrôle d'accès (ACL). Type : chaîne Valeur par défaut : private Valeurs valides : `private | public-read | public-read-write | aws-exec-read | authenticated-read | bucket-owner-read | bucket-owner-full-control`	Non
`Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires`	En-têtes spécifiques à REST. Pour de plus amples informations, veuillez consulter PUT Object.	Non
`key`	Nom de la clé chargée. Pour utiliser le nom de fichier fourni par l'utilisateur, utilisez la variable ${filename}. Par exemple, si l'utilisateur Betty charge le fichier lolcatz.jpg et que vous spécifiez /user/betty/${filename}, le fichier est stocké en tant que /user/betty/lolcatz.jpg. Pour de plus amples informations, veuillez consulter Utilisation des métadonnées d'objet.	Oui

Nom de champ	Description	Obligatoire
`policy`	Stratégie de sécurité décrivant ce qui est autorisé dans la demande. Les demandes sans stratégie de sécurité sont considérées anonymes et aboutiront uniquement sur des compartiments publiquement accessibles en écriture.	Non
`success_action_redirect, redirect`	URL vers laquelle le client est redirigé en cas d'échec du chargement. Amazon S3 ajoute à l'URL les valeurs de compartiment, de clé et etag comme paramètres de la chaîne d'interrogation. Si success_action_redirect n'est pas spécifié, Amazon S3 retourne le type de document vide spécifié dans le champ success_action_status. Si Amazon S3 ne peut pas interpréter l'URL, le champ est ignoré. Si le téléchargement échoue, Amazon S3 affiche une erreur et ne redirige pas l'utilisateur vers une URL. Pour de plus amples informations, veuillez consulter Redirection. **Note** Le nom du champ de redirection est obsolète et la prise en charge du nom du champ de redirection sera supprimée à l'avenir.	Non

Nom de champ	Description	Obligatoire
`success_action_status`	Code de statut retourné au client lors du succès du chargement si success_action_redirect n'est pas spécifié. Les valeurs valides sont 200, 201 et 204 (par défaut). Si la valeur est définie sur 200 ou 204, Amazon S3 retourne un document vide avec un code de statut égal à 200 ou 204. Si la valeur est définie sur 201, Amazon S3 retourne un document XML avec un code de statut égal à 201. Pour des informations sur le contenu du document XML, veuillez consulter POST Object. Si la valeur n'est pas définie ou si elle est définie sur une valeur non valide, Amazon S3 retourne un document vide avec un code de statut égal à 204. **Note** Certaines versions d'Adobe Flash Player ne traitent pas correctement les réponses HTTP dont le corps est vide. Pour prendre en charge les chargements via Adobe Flash, nous vous recommandons de définir `success_action_status` sur 201.	Non

Nom de champ	Description	Obligatoire
`signature`	La signature HMAC est construite à l'aide de la clé d'accès secrète qui correspond à celle fournieAWSAccessKeyId. Ce champ est requis si un document de stratégie est inclus dans la demande. Pour plus d'informations, consultez Identity and Access Management pour Amazon S3.	Conditionnel
`x-amz-security-token`	Jeton de sécurité utilisé par les informations d'identification de session Si la demande utilise Amazon DevPay , elle nécessite deux champs de `x-amz-security-token` formulaire : un pour le jeton de produit et un pour le jeton d'utilisateur. Si la demande utilise les informations d'identification de session, elle requiert un seul formulaire`x-amz-security-token` . Pour de plus amples informations, veuillez consulter Informations d'identification de sécurité temporaires dans le Guide de l'utilisateur IAM.	Non
Autres noms de champs préfixés par - x-amz-meta	Métadonnées spécifiées par l'utilisateur. Amazon S3 ne valide pas et n'utilise pas ces données. Pour de plus amples informations, veuillez consulter PUT Object.	Non

Nom de champ	Description	Obligatoire
dans le fichier	Fichier ou contenu de texte. Le fichier ou le contenu doit être le dernier champ du formulaire. Tout champ situé au-dessous est ignoré. Vous ne pouvez pas charger plus d'un fichier à la fois.	Oui

Elaboration de la stratégie

Rubriques

- Expiration
- Conditions
- Correspondance des conditions
- Échappement de caractère

La stratégie est un document JSON encodé en UTF-8 et Base64 qui spécifie les conditions que la demande doit respecter et qui est utilisé pour authentifier le contenu. En fonction de la manière dont vous concevez vos documents de stratégie, vous pouvez les utiliser par chargement, par utilisateur, pour tous les chargements ou selon d'autres conceptions répondant à vos besoins.

> **Note**
>
> Le document de stratégie est facultatif, mais nous recommandons vivement son utilisation plutôt que de rendre un compartiment publiquement accessible en écriture.

Voici un exemple de document de stratégie :

```
{ "expiration": "2007-12-01T12:00:00.000Z",

  "conditions": [
```

```
    {"acl": "public-read" },

    {"bucket": "awsexamplebucket1" },

    ["starts-with", "$key", "user/eric/"],

  ]

}
```

Le document de stratégie contient l'expiration et les conditions.

Expiration

L'élément expiration spécifie la date d'expiration de la stratégie au format de date UTC, conformément à la norme ISO 8601. Par exemple, « 2007-12-01T12:00:00.000Z » indique que la stratégie ne sera plus valide après minuit, heure UTC, le 01/12/2007. Une date d'expiration est requise dans une stratégie.

Conditions

Les conditions figurant dans le document de stratégie valident le contenu de l'objet chargé. Chaque champ de formulaire que vous spécifiez dans le formulaire (à l'exception des noms de signatureAWSAccessKeyId, de fichier, de politique et de champ dotés d'un préfixe x-ignore-) doit être inclus dans la liste des conditions.

> **Note**
>
> Si plusieurs champs ont le même nom, les valeurs doivent être séparées par des virgules. Par exemple, si vous avez deux champs nommés « x-amz-meta-tag » et que le premier a la valeur « Ninja » et le second la valeur « Stallman », vous devez définir le document de politique sur. `Ninja,Stallman`
>
> Toutes les variables figurant dans le formulaire sont développées avant la validation de la stratégie. Par conséquent, toutes les correspondances des conditions doivent être effectuées par rapport aux champs développés. Par exemple, si vous définissez le champ clé sur `user/betty/${filename}`, la stratégie peut être `[ "starts-with", "$key", "user/betty/" ]`. Ne saisissez pas `[ "starts-with", "$key", "user/betty/${filename}" ]`. Pour de plus amples informations, veuillez consulter Correspondance des conditions.

Le tableau ci-dessous décrit les conditions d'un document de stratégie.

Nom d'élément	Description
liste acl	Spécifie les conditions que la liste ACL doit respecter. Prend en charge la correspondance exacte et `starts-with` .
content-length-range	Spécifie les tailles minimale et maximale autorisées pour le contenu chargé. Prend en charge la correspondance de plage.
Cache-Control, Content-Type, Content-Disposition, Content-Encoding, Expires	En-têtes spécifiques à REST. Prend en charge la correspondance exacte et `starts-with` .
key	Nom de la clé chargée. Prend en charge la correspondance exacte et `starts-with` .
success_action_redirect, redirect	URL vers laquelle le client est redirigé en cas d'échec du chargement. Prend en charge la correspondance exacte et `starts-with` .
success_action_status	Code de statut retourné au client lors du succès du chargement si success_action_redirect n'est pas spécifié. Prend en charge la correspondance exacte.
x-amz-security-token	Jeton DevPay de sécurité Amazon.

Nom d'élément	Description
	Chaque demande qui utilise Amazon DevPay nécessite deux champs de `x-amz-security-token` formulaire : un pour le jeton de produit et un pour le jeton d'utilisateur. Par conséquent, les valeurs doivent être séparées par des virgules. Par exemple, si le jeton utilisateur est `eW91dHViZQ==` et le token produit `b0hnNVNKWVJIQTA=`, vous définissez l'entrée de stratégie sur : `{ "x-amz-security-token": "eW91dHViZQ==,b0hnNVNKWVJIQTA=" }`.
Autres noms de champs préfixés par - x-amz-meta	Métadonnées spécifiées par l'utilisateur. Prend en charge la correspondance exacte et `starts-with`.

> **Note**
>
> Si votre boîte à outils ajoute des champs supplémentaires (p. ex. : Flash ajoute le nom de fichier), vous devez les ajouter dans le document de stratégie. Si vous pouvez contrôler cette fonctionnalité, ajoutez le préfixe `x-ignore-` au champ afin qu'Amazon S3 ignore la fonction et que les versions futures de cette fonction ne soient pas affectées.

Correspondance des conditions

Le tableau ci-dessous décrit les types de correspondance des conditions. Vous devez spécifier une seule condition pour chaque champ de formulaire que vous spécifiez dans le formulaire, mais vous pouvez créer des critères de correspondance plus complexes en spécifiant plusieurs conditions pour un champ de formulaire.

Condition	Description

Condition	Description
Correspondances exactes	Les correspondances exactes vérifient que les champs correspondent à des valeurs spécifiques. Cet exemple indique que la liste ACL doit être définie sur public-read : `{"acl": "public-read" }` Cet exemple représente une méthode alternative pour indiquer que la liste ACL doit être définie sur public-read : `[ "eq", "$acl", "public-read" ]`
Commence par	Si la valeur doit commencer par une certaine valeur, utilisez le mot clé starts-with. Cet exemple indique que la clé doit commencer par user/betty : `["starts-with", "$key", "user/betty/"]`
Correspondance avec un contenu quelconque	Pour configurer la stratégie de manière à autoriser un contenu quelconque dans un champ, utilisez starts-with avec une valeur vide. Cet exemple autorise une valeur quelconque pour success_action_redirect : `["starts-with", "$success_action_redirect", ""]`
Spécification de plages	Pour les champs qui acceptent des plages, séparez les seuils inférieur et supérieur de plage par une virgule. Cet exemple autorise une taille de fichier comprise entre 1 et 10 mégaoctets : `["content-length-range", 1048579, 10485760]`

Échappement de caractère

Le tableau suivant décrit les caractères qui doivent être placés dans une séquence d'échappement au sein d'un document de stratégie.

Séquence d'échappement	Description
\\	Barre oblique inverse
\$	Symbole dollar
\b	Retour arrière
\f	Saut de page
\n	Nouvelle ligne
\r	Retour chariot
\t	Tabulation horizontale
\v	Tabulation verticale
\u*xxxx*	Tous les caractères Unicode

Élaboration d'une signature

Étape	Description
1	Encodez la stratégie en UTF-8.

Étape	Description
2	Encodez les octets UTF-8 en Base64.
3	Signez la stratégie avec votre clé d'accès secrète en utilisant HMAC SHA-1.
4	Encodez la signature SHA-1 en Base64.

Pour obtenir des informations générales sur l'authentification, veuillez consulter Identity and Access Management pour Amazon S3.

Redirection

Cette section décrit comment traiter les redirections.

Redirection générale

Une fois la demande POST terminée, l'utilisateur est redirigé à l'emplacement que vous avez spécifié dans le champ `success_action_redirect`. Si Amazon S3 ne peut pas interpréter l'URL, le champ `success_action_redirect` est ignoré.

Si `success_action_redirect` n'est pas spécifié, Amazon S3 retourne le type de document vide spécifié dans le champ `success_action_status`.

Si la demande POST échoue, Amazon S3 affiche une erreur et ne fournit pas de redirection.

Redirection antérieure au chargement

Si votre bucket a été créé à l'aide de < CreateBucketConfiguration >, il se peut que vos utilisateurs finaux aient besoin d'une redirection. Si cela se produit, certains navigateurs peuvent traiter la redirection de façon incorrecte. Cela est relativement rare, mais a le plus de chances de se produire juste après la création d'un compartiment.

Exemples de téléchargement (AWS signature version 2)

Rubriques

- Chargement d'un fichier
- Chargement d'une zone de texte

> **Note**
>
> L'authentification des demandes abordée dans cette section est basée sur AWS Signature Version 2, un protocole permettant d'authentifier les demandes d'API entrantes adressées aux AWS services.
> Amazon S3 prend désormais en charge la version 4 de Signature, un protocole permettant d'authentifier les demandes d'API entrantes adressées aux AWS services, dans l'ensemble. Régions AWSÀ l'heure actuelle, le protocole Régions AWS créé avant le 30 janvier 2014 continuera de prendre en charge le protocole précédent, Signature Version 2. Toutes les nouvelles Régions créées après le 30 janvier 2014 prendront en charge uniquement Signature Version 4. Par conséquent, toutes les demandes adressées à ces Régions doivent être effectuées avec Signature Version 4. Pour plus d'informations, consultez Exemples : téléchargement basé sur un navigateur à l'aide de HTTP POST (avec AWS signature version 4) dans le manuel Amazon Simple Storage Service API Reference.

Chargement d'un fichier

Cet exemple illustre le processus complet d'élaboration d'une stratégie et d'un formulaire pouvant être utilisés pour charger un fichier joint.

Élaboration d'une stratégie et d'un formulaire

La stratégie suivante prend en charge les chargements vers Amazon S3 pour le compartiment awsexamplebucket1.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
successful_upload.html"},
    ["starts-with", "$Content-Type", "image/"],
    {"x-amz-meta-uuid": "14365123651274"},
    ["starts-with", "$x-amz-meta-tag", ""]
  ]
}
```

Cette stratégie exige le respect des conditions suivantes :

- Le chargement doit se produire avant 12:00, heure UTC, le 1er décembre 2007.
- Le contenu doit être chargé dans le compartiment awsexamplebucket1.
- La clé doit commencer par « user/eric/ ».
- La liste ACL est définie sur public-read.
- La valeur success_action_redirect est définie sur https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html.
- L'objet est un fichier image.
- La x-amz-meta-uuid balise doit être définie sur 14365123651274.
- Ils x-amz-meta-tag peuvent contenir n'importe quelle valeur.

Voici une version encodée en Base64 de cette stratégie.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWN
```

Créez une signature à l'aide de vos informations d'identification. Par exemple, `0RavWzkygo6QX9caELEqKi9kDbU=` est la signature pour le document de stratégie précédent.

Le formulaire suivant prend en charge une demande POST vers le compartiment DOC-EXAMPLE-BUCKET qui utilise cette politique.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.amazonaws.com/" method="post"
 enctype="multipart/form-data">
    Key to upload: <input type="input" name="key" value="user/eric/" /><br />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="https://
awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html" />
    Content-Type: <input type="input" name="Content-Type" value="image/jpeg" /><br />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />
```

```
    <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />
    <input type="hidden" name="Policy" value="POLICY" />
    <input type="hidden" name="Signature" value="SIGNATURE" />
    File: <input type="file" name="file" /> <br />
    <!-- The elements after this will be ignored -->
    <input type="submit" name="submit" value="Upload to Amazon S3" />
  </form>
  ...
</html>
```

Exemple de demande

Cette demande suppose que l'image chargée a une taille de 117 108 octets. Les données de l'image ne sont pas incluses.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115
 Firefox/2.0.0.10
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: 118698

--9431149156168
Content-Disposition: form-data; name="key"

user/eric/MyPicture.jpg
--9431149156168
Content-Disposition: form-data; name="acl"

public-read
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

https://awsexamplebucket1.s3.us-west-1.amazonaws.com/successful_upload.html
--9431149156168
Content-Disposition: form-data; name="Content-Type"
```

```
image/jpeg
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

14365123651274
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

Some,Tag,For,Picture
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

AKIAIOSFODNN7EXAMPLE
--9431149156168
Content-Disposition: form-data; name="Policy"

eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWN
--9431149156168
Content-Disposition: form-data; name="Signature"

0RavWzkygo6QX9caELEqKi9kDbU=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg

...file content...
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to Amazon S3
--9431149156168--
```

Exemple de réponse

```
HTTP/1.1 303 Redirect
x-amz-request-id: 1AEE782442F35865
x-amz-id-2: cxzFLJRatFHy+NGtaDFRR8YvI9BHmgLxjvJzNiGGICARZ/mVXHj7T+qQKhdpzHFh
Content-Type: application/xml
Date: Wed, 14 Nov 2007 21:21:33 GMT
Connection: close
```

```
Location: https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
successful_upload.html?bucket=awsexamplebucket1&key=user/eric/
MyPicture.jpg&etag=&quot;39d459dfbc0faabbb5e179358dfb94c3&quot;
Server: AmazonS3
```

Chargement d'une zone de texte

Rubriques

- Élaboration d'une stratégie et d'un formulaire
- Exemple de demande
- Exemple de réponse

L'exemple suivant illustre le processus complet d'élaboration d'une stratégie et d'un formulaire pour charger une zone de texte. Le chargement d'une zone de texte est utile pour soumettre du contenu créé par l'utilisateur, tel que des billets de blog.

Élaboration d'une stratégie et d'un formulaire

La stratégie suivante prend en charge les chargements de zones de texte vers Amazon S3 pour le compartiment awsexamplebucket1.

```
{ "expiration": "2007-12-01T12:00:00.000Z",
  "conditions": [
    {"bucket": "awsexamplebucket1"},
    ["starts-with", "$key", "user/eric/"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://awsexamplebucket1.s3.us-west-1.amazonaws.com/
new_post.html"},
    ["eq", "$Content-Type", "text/html"],
    {"x-amz-meta-uuid": "14365123651274"},
    ["starts-with", "$x-amz-meta-tag", ""]
  ]
}
```

Cette stratégie exige le respect des conditions suivantes :

- Le chargement doit se produire avant 12:00, heure GMT, le 1er décembre 2007.
- Le contenu doit être chargé dans le compartiment awsexamplebucket1.
- La clé doit commencer par « user/eric/ ».

- La liste ACL est définie sur public-read.
- La valeur success_action_redirect est définie sur https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html.
- L'objet est un texte HTML.
- La x-amz-meta-uuid balise doit être définie sur 14365123651274.
- Ils x-amz-meta-tag peuvent contenir n'importe quelle valeur.

Voici une version encodée en Base64 de cette stratégie.

```
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXR
pb25zIjogWwogICAgeyJidWNrZXQiOiAiam9obnNtaXRoIn0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAidXNlci
LAogICAgeyJhY2wiOiAicHVibGljLXJlYWQifSwKICAgIHsic3VjY2Vzc19hY3Rpb25fcmVkaXJlY3QiOiAiaHR0cDovL2p
C5zMy5hbWF6b25hd3MuY29tL25ld19wb3N0Lmh0bWwifSwKICAgIFsiZXEiLCAiJENvbnRlbnQtVHlwZSIsICJ0ZXh0L2h0
CAgIHsieC1hbXotbWV0YS11dWlkIjogIjE0MzY1MTIzNjUxMjc0In0sCiAgICBbInN0YXJ0cy13aXRoIiwgIiR4LWFtei1t
IsICIiXQogIF0KfQo=
```

Créez une signature en utilisant vos informations d'identification. Par exemple, qA7FWXKq6VvU68lI9KdveT1cWgF= est la signature pour le document de stratégie précédent.

Le formulaire suivant prend en charge une demande POST vers le compartiment DOC-EXAMPLE-BUCKET qui utilise cette politique.

```
<html>
  <head>
    ...
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    ...
  </head>
  <body>
  ...
  <form action="https://DOC-EXAMPLE-BUCKET.s3.us-west-1.amazonaws.com/" method="post"
 enctype="multipart/form-data">
    Key to upload: <input type="input" name="key" value="user/eric/" /><br />
    <input type="hidden" name="acl" value="public-read" />
    <input type="hidden" name="success_action_redirect" value="https://
awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html" />
    <input type="hidden" name="Content-Type" value="text/html" />
    <input type="hidden" name="x-amz-meta-uuid" value="14365123651274" />
    Tags for File: <input type="input" name="x-amz-meta-tag" value="" /><br />
    <input type="hidden" name="AWSAccessKeyId" value="AKIAIOSFODNN7EXAMPLE" />
```

```
    <input type="hidden" name="Policy" value="POLICY" />
    <input type="hidden" name="Signature" value="SIGNATURE" />
    Entry: <textarea name="file" cols="60" rows="10">

Your blog post goes here.

    </textarea><br />
    <!-- The elements after this will be ignored -->
    <input type="submit" name="submit" value="Upload to Amazon S3" />
  </form>
  ...
</html>
```

Exemple de demande

Cette demande suppose que l'image chargée a une taille de 117 108 octets. Les données de l'image ne sont pas incluses.

```
POST / HTTP/1.1
Host: awsexamplebucket1.s3.us-west-1.amazonaws.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.10) Gecko/20071115
 Firefox/2.0.0.10
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=178521717625888
Content-Length: 118635

-178521717625888
Content-Disposition: form-data; name="key"

ser/eric/NewEntry.html
--178521717625888
Content-Disposition: form-data; name="acl"

public-read
--178521717625888
Content-Disposition: form-data; name="success_action_redirect"
```

```
https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html
--178521717625888
Content-Disposition: form-data; name="Content-Type"

text/html
--178521717625888
Content-Disposition: form-data; name="x-amz-meta-uuid"

14365123651274
--178521717625888
Content-Disposition: form-data; name="x-amz-meta-tag"

Interesting Post
--178521717625888
Content-Disposition: form-data; name="AWSAccessKeyId"

AKIAIOSFODNN7EXAMPLE
--178521717625888
Content-Disposition: form-data; name="Policy"
eyAiZXhwaXJhdGlvbiI6ICIyMDA3LTEyLTAxVDEyOjAwOjAwLjAwMFoiLAogICJjb25kaXRpb25zIjogWwogICAgeyJidWN

--178521717625888
Content-Disposition: form-data; name="Signature"

qA7FWXKq6VvU68lI9KdveT1cWgF=
--178521717625888
Content-Disposition: form-data; name="file"

...content goes here...
--178521717625888
Content-Disposition: form-data; name="submit"

Upload to Amazon S3
--178521717625888--
```

Exemple de réponse

```
HTTP/1.1 303 Redirect
x-amz-request-id: 1AEE782442F35865
x-amz-id-2: cxzFLJRatFHy+NGtaDFRR8YvI9BHmgLxjvJzNiGGICARZ/mVXHj7T+qQKhdpzHFh
Content-Type: application/xml
Date: Wed, 14 Nov 2007 21:21:33 GMT
Connection: close
```

```
Location: https://awsexamplebucket1.s3.us-west-1.amazonaws.com/new_post.html?
bucket=awsexamplebucket1&key=user/eric/
NewEntry.html&etag=40c3271af26b7f1672e41b8a274d28d4
Server: AmazonS3
```

POST avec Adobe Flash

Cette section décrit comment utiliser POST avec Adobe Flash.

Sécurité d'Adobe Flash Player

Par défaut, le modèle de sécurité d'Adobe Flash Player interdit aux lecteurs Adobe Flash Player d'établir des connexions réseau à des serveurs situés hors du domaine qui assure le service du fichier SWF.

Pour remplacer le paramètre par défaut, vous devez charger un fichier crossdomain.xml publiquement accessible en lecture dans le compartiment qui acceptera les chargements POST. Vous trouverez ci-dessous un exemple de fichier crossdomain.xml.

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*" secure="false" />
</cross-domain-policy>
```

> **Note**
>
> Pour plus d'informations sur le modèle de sécurité d'Adobe Flash, accédez au site web d'Adobe.
> L'ajout du fichier crossdomain.xml dans votre compartiment autorise tout lecteur Adobe Flash Player à se connecter au fichier crossdomain.xml au sein de votre compartiment. Toutefois, cela n'accorde pas l'accès au compartiment Amazon S3 lui-même.

Considérations sur Adobe Flash

L' FileReference API d'Adobe Flash ajoute le champ de `Filename` formulaire à la requête POST. Lorsque vous créez des applications Adobe Flash chargées sur Amazon S3 à l'aide de l'action d' FileReference API, incluez la condition suivante dans votre politique :

```
['starts-with', '$Filename', '']
```

Certaines versions d'Adobe Flash Player ne traitent pas correctement les réponses HTTP dont le corps est vide. Pour configurer POST de manière à retourner une réponse dont le corps n'est pas vide, définissez `success_action_status` sur 201. Amazon S3 retournera alors un document XML avec un code de statut égal à 201. Pour des informations sur le contenu du document XML, veuillez consulter POST Object. Pour obtenir des informations sur les champs de formulaire, consultez Champs de formulaire HTML.

Schémas de conception des bonnes pratiques : optimisation des performances Amazon S3

Vos applications peuvent facilement exécuter des milliers de transactions par seconde lors du chargement et de l'extraction du stockage depuis Amazon S3. Amazon S3 se met automatiquement à l'échelle en fonction des taux de demandes très importants. Par exemple, votre application peut atteindre au moins 3 500 demandes PUT/COPY/POST/DELETE ou 5 500 demandes GET/HEAD par seconde par préfixe Amazon S3 partitionné. Il n'existe aucune limite au nombre de préfixes dans un compartiment. Vous pouvez augmenter vos performances de lecture et d'écriture en effectuant une mise en parallèle. Par exemple, si vous créez 10 préfixes dans un compartiment Amazon S3 pour paralléliser les lectures, vous pouvez adapter vos performances de lecture à 55 000 demandes de lecture par seconde. De même, vous pouvez mettre à l'échelle les opérations d'écriture en écrivant sur plusieurs préfixes. La mise à l'échelle, dans le cas des opérations de lecture et d'écriture, se fait progressivement et n'est pas instantanée. Pendant la mise à l'échelle d'Amazon S3 à votre nouveau taux de demandes plus élevé, vous pouvez rencontrer des erreurs 503 (Ralentissement). Ces erreurs disparaissent une fois la mise à l'échelle terminée. Pour plus d'informations sur la création et l'utilisation de préfixes, consultez Organisation des objets à l'aide de préfixes.

Certaines applications de lacs de données sur Amazon S3 analysent des milliards d'objets pour les requêtes qui portent sur des péta-octets de données. Ces applications de lacs de données atteignent des taux de transfert à instance unique qui optimisent l'utilisation de l'interface réseau pour leur instance Amazon EC2, laquelle peut atteindre jusqu'à 100 Gbits/s sur une seule instance. Ces applications regroupent ensuite le débit de plusieurs instances pour parvenir à plusieurs téraoctets par seconde.

D'autres applications sont sensibles à la latence, comme les applications de messagerie des réseaux sociaux. Ces applications peuvent atteindre des latences constantes pour les petits objets (et les first-byte-out latences pour les objets plus grands) d'environ 100 à 200 millisecondes.

D'autres AWS services peuvent également contribuer à accélérer les performances de différentes architectures d'applications. Par exemple, si vous souhaitez des taux de transfert plus élevés sur une seule connexion HTTP ou des latences à un chiffre en millisecondes, utilisez Amazon CloudFront ou Amazon pour la mise en cache avec ElastiCacheAmazon S3.

De plus, si vous souhaitez un transport rapide de données sur de longues distances entre un client et un compartiment S3, utilisez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration. Transfer Acceleration utilise les emplacements périphériques

répartis dans le monde entier CloudFront pour accélérer le transport des données sur des distances géographiques. Si votre charge de travail Amazon S3 utilise le chiffrement côté serveur avec AWS KMS, consultez la section AWS KMS Limites du guide du AWS Key Management Service développeur pour obtenir des informations sur les taux de demandes pris en charge pour votre cas d'utilisation.

Les rubriques suivantes décrivent les instructions et les modèles de conception des bonnes pratiques pour optimiser les performances des applications qui utilisent Amazon S3. Veuillez consulter Recommandations de performance pour Amazon S3 et Modèles de conception des performances pour Amazon S3 pour obtenir les informations les plus à jour sur l'optimisation des performances pour Amazon S3.

> **Note**
>
> Pour plus d'informations sur l'utilisation de la classe de stockage Amazon S3 Express One Zone avec des compartiments de répertoires, consultez Qu'est-ce que S3 Express One Zone ? et Compartiments de répertoire.

Rubriques

- Recommandations de performance pour Amazon S3
- Modèles de conception des performances pour Amazon S3

Recommandations de performance pour Amazon S3

Lors du développement d'applications qui téléchargent et extraient les objets depuis Amazon S3, suivez nos instructions sur les bonnes pratiques pour optimiser les performances. Nous proposons également des plus détaillé Modèles de conception des performances.

Pour obtenir les meilleures performances pour votre application sur Amazon S3, nous recommandons les directives suivantes.

Rubriques

- Performances des mesures
- Mettre à l'échelle horizontalement les connexions de stockage

- Utiliser les extractions de plages d’octets
- Nouvelle tentative de demandes pour les applications sensibles à la latence
- Combinez Amazon S3 (stockage) et Amazon EC2 (calcul) dans le même environnement Région AWS
- Utiliser Amazon S3 Transfer Acceleration pour réduire la latence provoquée par la distance
- Utiliser la version la plus récente des kits SDK AWS

Performances des mesures

Lors de l’optimisation des performances, examinez le débit du réseau, l’UC et les exigences DRAM. Selon les mélange des demandes de ces différentes ressources, il peut valoir la peine d'évaluer différents types d'instance Amazon EC2. Pour plus d'informations sur les types d'instances, consultez la section Types d'instances dans le guide de l'utilisateur Amazon EC2.

Il peut aussi être utile d’examiner le temps de recherche DNS, la latence et la vitesse de transfert des données à l’aide des outils d’analyse HTTP lors de la mesure des performances.

Pour comprendre les exigences de performances et optimiser les performances de votre application, vous pouvez également surveiller les réponses d'erreur 503 que vous recevez. La surveillance de certaines métriques de performances peut entraîner des dépenses supplémentaires. Pour plus d’informations, consultez Tarification Amazon S3.

Surveillance du nombre de réponses d'erreur de statut 503 (Ralentissement)

Pour surveiller le nombre de réponses d'erreur de statut 503 que vous recevez, vous pouvez utiliser l'une des options suivantes :

- Utilisez les métriques de CloudWatch demande Amazon pour Amazon S3. Les métriques de CloudWatch demande incluent une métrique pour les réponses d'état 5xx. Pour plus d'informations sur les métriques relatives aux CloudWatch demandes, consultezSurveillance des métriques avec Amazon CloudWatch.
- Utilisez le nombre d'erreurs 503 (Service non disponible) disponible dans la section des métriques avancées d'Amazon S3 Storage Lens. Pour plus d’informations, consultez Utilisation de métriques S3 Storage Lens pour améliorer les performances.
- Utilisez la journalisation des accès au serveur Amazon S3. La journalisation des accès au serveur vous permet de filtrer et de passer en revue toutes les demandes qui reçoivent des réponses 503 (Erreur interne). Vous pouvez également utiliser Amazon Athena pour analyser les journaux. Pour

en savoir plus sur la journalisation des accès au serveur, consultez Enregistrement de demandes avec journalisation des accès au serveur.

En surveillant le nombre de codes d'erreur de statut HTTP 503, vous pouvez souvent obtenir des insights précieux sur les préfixes, les clés ou les compartiments qui reçoivent le plus de demandes de limitation.

Mettre à l’échelle horizontalement les connexions de stockage

La répartition des demandes entre plusieurs connexions est un modèle de conception courant pour mettre horizontalement à l’échelle les performances. Lorsque vous développez des applications hautes performances, pensez à Amazon S3 comme à un très grand système réparti, et non comme un simple point de terminaison réseau, tel qu'un serveur de stockage traditionnel. Vous pouvez atteindre les meilleures performances en adressant plusieurs demandes concurrentes à Amazon S3. Répartissez ces demandes sur des connexions distinctes pour maximiser la bande passante accessible à partir d'Amazon S3. Amazon S3 n'a aucune limite pour le nombre de connexions à votre compartiment.

Utiliser les extractions de plages d’octets

Avec l'en-tête HTTP `Range` dans une demande GetObject, vous pouvez extraire une plage d'octets d'un objet, en ne transférant que la portion spécifiée. Vous pouvez utiliser des connexions simultanées vers Amazon S3 pour extraire différentes plages d'octets depuis le même objet. Vous pouvez ainsi parvenir au regroupement de débits le plus élevé, par opposition à une seule demande d’objet entier. L’extraction des plages les plus petites d’un grand objet permet aussi à votre application d’améliorer l’intervalle des nouvelles tentatives quand les demandes sont interrompues. Pour plus d’informations, consultez Téléchargement d'objets.

Les tailles traditionnelles des demandes de plages d’octets sont de 8 Mo ou 16 Mo. Si les objets sont l’objet d’une opération PUT à l’aide d’un chargement en plusieurs parties, une bonne pratique consiste à les soumettre à une opération GET dans les mêmes tailles d’élément (ou au moins alignées sur les frontières d’élément) pour obtenir de meilleures performances. Les demandes GET peuvent directement adresser les éléments individuels ; par exemple,, `GET ?partNumber=N.`

Nouvelle tentative de demandes pour les applications sensibles à la latence

Les expirations et les nouvelles tentatives agressives permettent d’obtenir une latence cohérente. Étant donnée la large échelle d'Amazon S3, si la première demande est lente, une demande de

nouvelle tentative est susceptible d'emprunter un chemin différent et de réussir rapidement. Les AWS SDK ont des valeurs de délai d'expiration et de nouvelle tentative configurables que vous pouvez ajuster en fonction des tolérances de votre application spécifique.

Combinez Amazon S3 (stockage) et Amazon EC2 (calcul) dans le même environnement Région AWS

Même si les noms de compartiment S3 sont globalement uniques, chaque compartiment est stocké dans une région que vous sélectionnez lorsque vous créez le compartiment. Pour optimiser les performances, nous vous recommandons d'accéder au compartiment à partir des instances Amazon EC2 de la même manière Région AWS lorsque cela est possible. Cela permet de réduire la latence réseau et les coûts de transfert des données.

Pour plus d'informations sur la tarification du transfert des données, consultez la Tarification Amazon S3.

Utiliser Amazon S3 Transfer Acceleration pour réduire la latence provoquée par la distance

Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration permet un transfert rapide, facile et sécurisé des fichiers sur des longues distances entre votre client et un compartiment S3. Transfer Acceleration tire parti des emplacements périphériques répartis dans le monde entier sur Amazon CloudFront. Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé. Transfer Acceleration convient parfaitement au transfert régulier de gigaoctets ou téraoctets de données d'un continent à l'autre. Il est aussi utile pour les clients qui chargent sur un compartiment centralisé depuis le monde entier.

Vous pouvez utiliser l'outil de comparaison de vitesse Amazon S3 Transfer Acceleration pour comparer les vitesses de chargement accéléré et non accéléré dans les régions Amazon S3. L'outil de comparaison de la vitesse utilise les chargements partitionnés pour transférer un fichier à partir de votre navigateur vers différentes régions Amazon S3 avec ou sans Amazon S3 Transfer Acceleration.

Utiliser la version la plus récente des kits SDK AWS

Les AWS SDK fournissent une prise en charge intégrée de nombreuses directives recommandées pour optimiser les performances d'Amazon S3. Les kits SDK fournissent une API plus simple pour tirer parti d'Amazon S3 depuis une application. Ils sont régulièrement mis à jour pour suivre

les bonnes pratiques les plus récentes. Par exemple, les kits SDK incluent la logique permettant automatiquement une demande de nouvelle tentative sur les erreurs HTTP 503 et examinent le code pour répondre aux connexions lentes et s'y adapter.

Les kits SDK fournissent aussi Transfer Manager, qui automatise la mise à l'échelle horizontale des connexions pour traiter des milliers de demandes par seconde, à l'aide de demandes de plages d'octets si approprié. Il est important d'utiliser la dernière version des AWS SDK pour obtenir les dernières fonctionnalités d'optimisation des performances.

Vous pouvez aussi optimiser les performances lorsque vous utilisez les demandes d'API REST HTTP. Lors de l'utilisation de l'API REST, vous devez suivre les mêmes bonnes pratiques qui font partie des kits SDK. Autorisez les délais d'expiration et les nouvelles tentatives sur les demandes lentes, et multipliez les connexions pour autoriser l'extraction de données d'objet en parallèle. Pour plus d'informations sur l'utilisation de l'API REST, consultez la Référence d'API Amazon Simple Storage Service.

Modèles de conception des performances pour Amazon S3

Lors de la conception d'applications pour télécharger et extraire des objets depuis Amazon S3, utilisez nos modèles de conception des bonnes pratiques pour parvenir aux meilleures performances de votre application. Nous vous proposons aussi de prendre en compte des Instructions sur les performances lors de la planification de l'architecture de votre application.

Pour optimiser les performances, vous pouvez utiliser les modèles de conception suivants.

Rubriques

Utilisation de la mise en cache pour le contenu accédé fréquemment

La plupart des applications qui stockent les données dans Amazon S3 traitent un « ensemble opérationnel » de données, demandé à maintes reprises par les utilisateurs. Si une charge de travail envoie des requêtes GET répétées pour un ensemble commun d'objets, vous pouvez utiliser un

cache tel qu'Amazon CloudFront ElastiCache, Amazon ou AWS Elemental MediaStorepour optimiser les performances. L'adoption réussie du cache peut se traduire par une latence faible et des taux élevés de transfert des données. Les applications qui utilisent la mise en cache envoient aussi moins de demandes directes à Amazon S3, ce qui peut aider à réduire les coûts des demandes.

Amazon CloudFront est un réseau de diffusion de contenu rapide (CDN) qui met en cache de manière transparente les données d'Amazon S3 dans un grand nombre de points de présence répartis géographiquement (). PoPs Lorsque des objets sont accessibles depuis plusieurs régions ou via Internet, cela CloudFront permet aux données d'être mises en cache à proximité des utilisateurs qui accèdent aux objets. Il peut en résulter une diffusion haute performance des contenus Amazon S3 réputés. Pour plus d'informations à ce sujet CloudFront, consultez le manuel Amazon CloudFront Developer Guide.

Amazon ElastiCache est un cache en mémoire géré. Avec ElastiCache, vous pouvez provisionner des instances Amazon EC2 qui mettent en cache des objets en mémoire. Cette mise en cache se traduit par des ordres de réduction de magnitude dans la latence GET et de substantielles augmentations dans le débit de téléchargement. Pour l'utiliser ElastiCache, vous modifiez la logique de l'application pour remplir le cache avec des objets chauds et vérifier la présence d'objets chauds dans le cache avant de les demander à Amazon S3. Pour des exemples d'utilisation visant ElastiCache à améliorer les performances GET d'Amazon S3, consultez le billet de blog Turbocharger Amazon S3 avec Amazon ElastiCache pour Redis.

AWS Elemental MediaStore est un système de mise en cache et de distribution de contenu spécialement conçu pour les flux de travail vidéo et la diffusion de contenu multimédia à partir d'Amazon S3. MediaStore fournit des API end-to-end de stockage spécifiques pour la vidéo et est recommandé pour les charges de travail vidéo sensibles aux performances. Pour plus d'informations à ce sujet MediaStore, consultez le guide de AWS Elemental MediaStore l'utilisateur.

Délais d'expiration et nouvelles tentatives pour les applications sensibles à la latence

Dans certaines situations, une application reçoit une réponse d'Amazon S3 indiquant qu'une nouvelle tentative est nécessaire. Amazon S3 mappe les noms de compartiment et d'objet aux données d'objet qui leur sont associées. Si une application génère des taux de demande élevés (généralement des taux soutenus de plus de 5 000 demandes par seconde pour un petit nombre d'objets), elle peut recevoir des réponses HTTP 503 slowdown. Si ces erreurs se produisent, chaque kit SDK AWS implémente une logique de nouvelle tentative automatique à l'aide d'une interruption exponentielle. Si vous n'utilisez pas de kit SDK AWS , vous devez implémenter une logique de nouvelle tentative lors

de la réception de l'erreur HTTP 503. Pour plus d'informations sur les techniques de rétrogradation, voir Rétentatives d'erreur et régression exponentielle dans le. AWSRéférence générale d'Amazon Web Services

Amazon S3 se met automatiquement à l'échelle en réponse aux taux soutenus de nouvelles demandes, optimisant dynamiquement les performances. Tandis qu'Amazon S3 optimise en interne le taux de nouvelles demandes, vous recevez temporairement des réponses HTTP 503 jusqu'à ce que l'optimisation soit terminée. Après qu'Amazon S3 optimise en interne les performances pour le nouveau taux de demandes, toutes les demandes sont généralement traitées sans nouvelles tentatives.

Pour les applications sensibles à la latence, Amazon S3 conseille un suivi et des opérations de nouvelles tentatives plus lentes. Lorsque vous retentez une demande, nous recommandons l'utilisation d'une nouvelle connexion à Amazon S3 et l'exécution d'une nouvelle recherche DNS.

Lorsque vous effectuez des demandes volumineuses de taille variable (par exemple, plus de 128 Mo), nous conseillons de suivre le débit obtenu et de procéder à de nouvelles tentatives sur les 5 % les plus lents des demandes. Lorsque vous exécutez des demandes plus petites (par exemple, inférieures à 512 Ko), où les latences médianes sont souvent dans une plage de dix millisecondes, une bonne instruction consiste à retenter une opération GET ou PUT après 2 secondes. Si de nouvelles tentatives sont nécessaires, la bonne pratique consiste à se retirer. Par exemple, nous recommandons de n'émettre une nouvelle tentative qu'après 2 secondes et une deuxième nouvelle tentative au bout de 4 secondes supplémentaires.

Si votre application adresse des demandes de taille fixe à Amazon S3, vous devez escompter des temps de réponse plus cohérents pour chacune de ces demandes. Dans ce cas, une stratégie simple consiste à identifier le 1 % de delandes les plus lentes et de les réessayer. Même une simple nouvelle tentative est fréquemment efficace pour réduire la latence.

Si vous utilisez AWS Key Management Service (AWS KMS) pour le chiffrement côté serveur, consultez la section Limites du manuel du AWS Key Management Service développeur pour obtenir des informations sur les taux de demandes pris en charge pour votre cas d'utilisation.

Mise à l'échelle horizontale et mise en parallèle des demandes pour le haut débit

Amazon S3 est un système distribué très large. Pour vous aider à tirer parti de son échelle, nous vous encourageons à horizontalement mettre à l'échelle les demandes parallèles adressées aux

points de terminaison du service Amazon S3. En plus de la distribution des demandes au sein d'Amazon S3, ce type d'approche de la mise à l'échelle permet de répartir la charge sur plusieurs chemins d'accès du réseau.

Pour les transferts à haut débit, Amazon S3 conseille l'utilisation d'applications employant plusieurs connexions pour exécuter des opérations GET ou PUT sur les données en parallèle. Par exemple, cela est pris en charge par Amazon S3 Transfer Manager dans le SDK AWS Java, et la plupart des autres AWS SDK proposent des structures similaires. Pour certaines applications, vous pouvez obtenir des connexions parallèles en lançant plusieurs demandes simultanément dans différents threads d'application ou dans différentes instances d'application. La meilleure approche à adopter dépend de votre application et de la structure des objets auxquels vous accédez.

Vous pouvez utiliser les AWS SDK pour émettre directement des requêtes GET et PUT plutôt que d'utiliser la gestion des transferts dans le AWS SDK. Cette approche vous permet d'affiner votre charge de travail plus directement, tout en tirant profit du support du kit SDK pour les nouvelles tentatives et sa gestions des réponses HTTP 503 qui peuvent se produire. En règle générale, lorsque vous téléchargez des objets volumineux au sein d'une région depuis Amazon S3 vers Amazon EC2, nous vous suggérons d'effectuer des demandes simultanées pour les plages d'octets d'un objet avec la granularité de 8 à 16 Mo. Effectuez une seule demande simultanée pour chaque 85-90 Mbits/s du débit réseau souhaité. Pour saturer une carte NIC 10 Gbits/s, vous pouvez utiliser jusqu'à 15 demandes simultanées sur des connexions distinctes. Vous pouvez augmenter les demandes simultanées sur plusieurs connexions pour saturer les NIC les plus rapides, telles que les NIC 25 Gbits/s ou 100 Gbits/s.

La mesure des performances est importante quand vous réglez le nombre de demandes à émettre simultanément. Il est recommandé de commencer avec une seule demande à la fois. Mesurez la bande passante réseau obtenue et l'utilisation des autres ressources que votre application utilise dans le traitement des données. Vous pouvez alors identifier la ressource du goulet d'étranglement (à savoir, la ressource avec l'utilisation la plus élevée), et de là le nombre de requêtes suscpetibles d'être utiles. Par exemple, si le traitement d'une demande à la fois conduit à une utilisation de l'UC de 25 %, cela induit que quatre demandes simultanées au plus peuvent être accueillies. Les mesures sont essentielles et il vaut la peine de confirmer l'utilisation des ressources tandis que le taux des demandes augmente.

Si votre application adresse directement des demandes à Amazon S3 à l'aide de l'API REST, nous recommandons l'utilisation d'un groupe de connexions HTTP et la réutilisation de chaque connexion pour un ensemble de demandes. Le fait d'éviter la configuration des connexions par demande supprime la nécessité d'exécuter des liaisons TCP à démarrage lent et SSL (Secure Sockets Layer)

sur chaque demande. Pour plus d'informations sur l'utilisation de l'API REST, consultez la Référence d'API Amazon Simple Storage Service.

Enfin, il vaut la peine de prêter attention à DNS et de vérifier que les demandes sont réparties sur un vaste groupe d'adresses IP Amazon S3. Les requêtes DNS pour le Amazon S3 parcourent une large liste de points de terminaison IP. Mais la mise en cache des programmes de résolution ou du code d'application qui réutilise une seule adresse IP ne tire pas parti de la diversité des adresses et de l'équilibrage de charge qui en découle. Les outils d'utilitaire réseau comme l'outil de ligne de commande `netstat` peuvent afficher les adresses IP utilisées pour la communication avec Amazon S3 et nous fournissons des instructions sur les configurations DNS à utiliser. Pour plus d'informations sur ces instructions, consultez Demandes.

Utilisation d'Amazon S3 Transfer Acceleration pour accélérer les transferts de données disparates géographiquement

Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration est efficace pour réduire ou supprimer la latence provoquée par la distance géographique entre les clients dispersés géographiquement et une application régionale qui utilise Amazon S3. Transfer Acceleration utilise les emplacements périphériques répartis dans le monde entier CloudFront pour le transport des données. Le réseau AWS périphérique possède des points de présence dans plus de 50 sites. Aujourd'hui, il est utilisé pour distribuer du contenu via Amazon Route 53 CloudFront et pour fournir des réponses rapides aux requêtes DNS adressées à Amazon Route 53.

Le réseau périphérique aide également à accélérer les transferts de données dans et en dehors d'Amazon S3. Il convient parfaitement aux applications qui transfèrent les données entre les continents, ont une connexion Internet rapide, utilise des objets de grande taille ou ont un contenu volumineux à charger. Lorsque les données arrivent dans un emplacement périphérique, elles sont transférées vers Amazon S3 sur un chemin de réseau optimisé. En général, plus vous êtes loin d'une région Amazon S3, plus vous pouvez escompter une amélioration de la vitesse grâce à Transfer Acceleration.

Vous pouvez configurer Transfer Acceleration sur les compartiments nouveaux ou existants. Vous pouvez utiliser un point de terminaison Amazon S3 Transfer Acceleration distinct pour utiliser les emplacements AWS périphériques. Le meilleur moyen de tester si Transfer Acceleration améliore les performances des demandes des clients consiste à utiliser l'Outil de comparaison de la vitesse de Amazon S3 Transfer Acceleration. Les configurations et les états du réseau varient de temps à autre et d'emplacement à emplacement. Par conséquent, vous n'êtes facturé que pour les

transferts où Amazon S3 Transfer Acceleration peut potentiellement améliorer vos performances de chargement. Pour plus d'informations sur l'utilisation de Transfer Acceleration avec différents AWS SDK, consultezActivation et utilisation de S3 Transfer Acceleration.

Qu'est-ce que Amazon S3 sur Outposts ?

AWS Outposts est un service entièrement géré qui offre la même AWS infrastructure, les mêmes AWS services, les mêmes API et les mêmes outils à pratiquement tous les centres de données, espaces de colocation ou installations sur site pour une expérience hybride véritablement cohérente. AWS Outposts est idéal pour les charges de travail qui nécessitent un accès à faible latence aux systèmes sur site, le traitement local des données, la résidence des données et la migration d'applications avec des interdépendances entre les systèmes locaux. Pour plus d'informations, consultez Présentation d' AWS Outposts dans le Guide de l'utilisateur AWS Outposts .

Avec Amazon S3 sur Outposts, vous pouvez créer des compartiments S3 sur vos Outposts afin de stocker et récupérer facilement des objets sur site. S3 sur Outposts fournit une nouvelle classe de stockage, `OUTPOSTS`, qui utilise les API Amazon S3 et est conçue pour stocker les données de manière durable et redondante sur plusieurs appareils et serveurs de vos Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC).

Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outposts que sur Simple Storage Service (Amazon S3), telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts via l'API AWS Management Console, AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST.

Comment fonctionne S3 sur Outposts

S3 sur Outposts est un service de stockage d'objets qui stocke les données sous forme d'objets dans des compartiments sur votre Outpost. Un objet est un fichier et toutes les métadonnées qui le décrivent. Un compartiment est un conteneur d'objets.

Pour stocker vos données dans S3 sur Outposts, vous devez d'abord créer un compartiment. Lorsque vous créez le compartiment, vous spécifiez un nom de compartiment et l'Outpost qui va contenir le compartiment. Pour accéder à votre compartiment S3 sur Outposts et effectuer des opérations sur les objets, vous devez ensuite créer et configurer un point d'accès. Vous devez également créer un point de terminaison pour router les requêtes vers votre point d'accès.

Les points d'accès simplifient l'accès aux données pour Service AWS toute application cliente qui stocke des données dans S3. Les points d'accès sont des points de terminaison réseau nommés qui sont attachés aux compartiments et peuvent être utilisés pour effectuer des opérations sur les objets, telles que `GetObject` et `PutObject`. Chaque point d'accès dispose d'autorisations et de contrôles réseau distincts.

Vous pouvez créer et gérer vos compartiments, points d'accès et points de terminaison S3 on Outposts à l'aide des AWS SDK ou de l' AWS Management Console API AWS CLI REST. Pour télécharger et gérer des objets dans votre compartiment S3 on Outposts, vous pouvez utiliser les AWS SDK ou l' AWS CLI API REST.

Régions

Pendant le AWS Outposts provisionnement, vous créez ou AWS créez une connexion de liaison de service qui reconnecte votre Outpost à la région de votre choix ou à la région d'origine de l' Région AWS Outpost pour les opérations de bucket et la télémétrie. Un Outpost repose sur la connectivité avec le parent Région AWS. Le rack Outposts n'est pas conçu pour les opérations déconnectées ou les environnements présentant une connectivité limitée ou nulle. Pour obtenir plus d'informations, consultez la section Outpost connectivity to Régions AWS dans le Guide d'utilisateur AWS Outposts .

Compartiments

Un compartiment est un conteneur pour les objets stockés dans S3 sur Outposts. Vous pouvez stocker un nombre quelconque d'objets dans un compartiment et vous pouvez avoir jusqu'à 100 compartiments par compte et par Outpost.

Lorsque vous créez un compartiment, vous saisissez un nom de compartiment et sélectionnez l'Outpost où le compartiment sera hébergé. Après avoir créé un compartiment, vous ne pouvez pas changer le nom du compartiment ou le déplacer vers un autre Outpost. Les noms de compartiments doivent suivre les règles de dénomination de compartiment Amazon S3. Dans S3 on Outposts, les noms de bucket sont propres à un Outpost et. Compte AWS Les compartiments S3 sur Outposts nécessitent `outpost-id`, `account-id` et le nom du compartiment pour les identifier.

L'exemple suivant montre le format Amazon Resource Name (ARN) pour les compartiments S3 sur Outposts. L'ARN est composé de la région où se trouve votre Outpost, de votre compte Outpost, de l'ID de l'Outpost et du nom du compartiment.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'ARN du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN du point d'accès pour S3 sur Outposts, qui inclut l'`outpost-id`, l'`account-id` et le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les compartiments, consultez Utilisation des compartiments S3 on Outposts.

Objets

Les objets sont les entités fondamentales stockées dans S3 sur Outposts. Les objets sont composés de données et de métadonnées. Les métadonnées sont un ensemble de paires nom-valeur décrivant des objets. Ces pairs comprennent certaines métadonnées par défaut telles que la date de la dernière modification et des métadonnées HTTP standard comme `Content-Type`. Vous pouvez aussi spécifier des métadonnées personnalisées au moment du stockage de l'objet. Un objet est identifié de manière unique dans un compartiment par une clé (ou un nom).

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Lorsque vous AWS installez un rack Outpost, vos données restent locales dans votre Outpost afin de répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme il AWS Management Console est hébergé dans la région, vous ne pouvez pas utiliser la console pour télécharger ou gérer des objets dans votre Outpost. Cependant, vous pouvez utiliser l'API REST AWS Command Line Interface (AWS CLI) et AWS les SDK pour télécharger et gérer vos objets via vos points d'accès.

Clés

Une clé d'objet (ou nom de clé) est l'identifiant unique d'un objet au sein d'un compartiment. Chaque objet d'un compartiment possède une clé et une seule. La combinaison d'un compartiment, d'une clé et d'un ID de version identifie chaque objet de manière unique.

L'exemple suivant montre le format ARN pour les objets S3 on Outposts, qui inclut le Région AWS code de la région dans laquelle l'Outpost est hébergé, l'ID, l' Compte AWS ID de l'Outpost, le nom du bucket et la clé d'objet :

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/
bucket/example-s3-bucket1/object/myobject
```

Pour en savoir plus sur les clés d'objet, consultez Utilisation des objets S3 on Outposts.

Gestion des versions S3

Vous pouvez utiliser la gestion des versions S3 sur des compartiments Outposts pour conserver plusieurs variantes d'un objet dans le même compartiment. Vous pouvez utiliser la gestion des versions S3 pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans vos compartiments. La gestion des versions S3 vous aide à récupérer en cas d'action involontaire d'un utilisateur et de défaillance applicative.

Pour plus d'informations, consultez Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.

ID de version

Lorsque vous activez la gestion des versions S3 pour un compartiment, S3 sur Outposts génère un ID de version unique pour chaque objet ajouté au compartiment. Les objets qui existaient déjà dans le compartiment au moment où vous activez la gestion des versions ont un ID de version égal à `null`. Si vous modifiez ces objets (ou tout autre) par d'autres opérations, par exemple PutObject, les nouveaux objets obtiennent un ID de version unique.

Pour plus d'informations, consultez Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.

Classe de stockage et chiffrement

S3 sur Outposts fournit une nouvelle classe de stockage, S3 Outposts (`OUTPOSTS`). La classe de stockage S3 Outposts n'est disponible que pour les objets stockés dans des compartiments sur AWS Outposts. Si vous essayez d'utiliser d'autres classes de stockage S3 avec S3 sur Outposts, celui-ci renvoie l'erreur `InvalidStorageClass`.

Par défaut, les objets stockés dans la classe de stockage S3 Outposts (`OUTPOSTS`) sont toujours chiffrés à l'aide du chiffrement côté serveur, avec les clés de chiffrement gérées par Amazon S3 (SSE-S3). Pour plus d'informations, consultez Chiffrement des données dans S3 on Outposts.

Politique de compartiment

Une politique de compartiment est une politique basée sur les ressources AWS Identity and Access Management (IAM) que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko.

Les stratégies de compartiment utilisent le langage de politique IAM basé sur JSON standard dans AWS. Vous pouvez utiliser des stratégies de compartiment pour ajouter ou refuser des autorisations pour les objets d'un compartiment. Les stratégies de compartiment autorisent ou refusent les requêtes en fonction des éléments de la stratégie. Ces éléments peuvent comprendre le demandeur, les actions de S3 sur Outposts, les ressources et les aspects ou conditions de la requête (par exemple, l'adresse IP utilisée pour effectuer la requête). Par exemple, vous pouvez créer une politique de compartiment qui accorde des autorisations inter-comptes pour charger des objets dans un compartiment S3 sur Outposts, tout en veillant à ce que le propriétaire du compartiment ait le contrôle total des objets chargés. Pour plus d'informations, consultez Exemples de politiques relatives aux compartiments Amazon S3.

Dans votre politique de compartiment, vous pouvez utiliser des caractères génériques (`*`) dans les ARN et d'autres valeurs pour accorder des autorisations à un sous-ensemble d'objets. Par exemple, vous pouvez contrôler l'accès aux groupes d'objets qui commencent par un préfixe courant ou se terminent par une extension donnée, comme `.html`.

Points d'accès S3 sur Outposts

Les points d'accès S3 sur Outposts sont des points de terminaison réseau nommés avec des stratégies d'accès dédiées qui décrivent la manière d'accéder aux données en utilisant ce point de terminaison. Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les jeux de données partagés dans S3 sur Outposts. Les points d'accès sont rattachés à des compartiments que vous pouvez utiliser pour effectuer des opérations sur des objets S3, telles que `GetObject` et `PutObject`.

Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'ARN du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

Les points d'accès disposent d'autorisations et de contrôles de réseau distincts que S3 on Outposts applique pour toute requête effectuée via ce point d'accès. Chaque point d'accès applique une stratégie de point d'accès personnalisée qui fonctionne conjointement avec la stratégie de compartiment associée au compartiment sous-jacent.

Pour plus d'informations, consultez Accès à vos compartiments et objets S3 on Outposts.

Caractéristiques de S3 sur Outposts

Gestion des accès

S3 sur Outposts offre des fonctionnalités d'audit et de gestion de l'accès à vos compartiments et objets. Par défaut, les compartiments S3 sur Outposts et les objets qu'ils contiennent sont privés. Vous n'avez accès qu'aux ressources S3 sur Outposts que vous créez.

Pour accorder des autorisations de ressources granulaires qui prennent en charge votre cas d'utilisation spécifique ou pour auditer les autorisations de vos ressources S3 sur Outposts, vous pouvez utiliser les fonctionnalités suivantes.

- S3 Block Public Access (Bloquer l'accès public S3) — bloquer l'accès public des compartiments S3 et des objets. Pour les compartiments sur les Outposts, le blocage de l'accès public est toujours activé par défaut.
- AWS Identity and Access Management (IAM) — IAM est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources, y compris à vos ressources S3 on Outposts. Avec IAM, vous pouvez gérer de manière centralisée les autorisations qui contrôlent

les ressources AWS auxquelles les utilisateurs peuvent accéder. Vous pouvez utiliser IAM pour contrôler les personnes qui s'authentifient (sont connectées) et sont autorisées (disposent d'autorisations) à utiliser des ressources.

- S3 on Outposts access points (Points d'accès S3 sur Outposts) — Gérez l'accès aux données pour les jeux de données partagés dans S3 sur Outposts. Les points d'accès sont nommés points de terminaison réseau avec des stratégies d'accès dédiées. Les points d'accès sont attachés aux compartiments et peuvent être utilisés pour effectuer des opérations sur les objets, par exemple `GetObject` et `PutObject`.
- Politiques de compartiment – Utilisez un langage de politique basé sur IAM pour configurer les autorisations basées sur les ressources de vos compartiments S3 et des objets qu'ils contiennent.
- AWS Resource Access Manager (AWS RAM) — Partagez en toute sécurité votre capacité S3 on Outposts entre Comptes AWS, au sein de votre organisation ou de vos unités organisationnelles (UO) dans. AWS Organizations

Journalisation et surveillance du stockage

S3 sur Outposts fournit des outils de journalisation et de surveillance que vous pouvez utiliser pour surveiller et contrôler l'utilisation de vos ressources S3 sur Outposts. Pour plus d'informations, consultez Outils de surveillance.

- Amazon CloudWatch Metrics for S3 on Outposts : suivez l'état de fonctionnement de vos ressources et déterminez la disponibilité de vos capacités.
- CloudWatch Événements Amazon Events pour S3 on Outposts — Créez une règle pour tout événement d'API S3 on Outposts afin de recevoir des notifications via toutes les cibles d'CloudWatch événements prises en charge, notamment Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) et. AWS Lambda
- AWS CloudTrail journaux pour S3 sur Outposts — Enregistrez les actions effectuées par un utilisateur, un rôle ou une action Service AWS dans S3 sur Outposts. CloudTrail les journaux vous fournissent un suivi détaillé des API pour les opérations au niveau du bucket S3 et au niveau de l'objet.

Forte cohérence

S3 on Outposts fournit une forte read-after-write cohérence pour les requêtes PUT et DELETE des objets de votre bucket S3 on Outposts dans l'ensemble. Régions AWS Ce comportement s'applique

à la fois aux écritures de nouveaux objets, aux requêtes PUT qui écrasent des objets existants et aux requêtes DELETE. En outre, les balises des objets S3 sur Outposts et les métadonnées des objets (par exemple, l'objet HEAD) sont fortement cohérentes. Pour plus d'informations, consultez Modèle de cohérence des données Amazon S3.

Services connexes

Une fois que vous avez chargé vos données dans S3 sur Outposts, vous pouvez les utiliser avec d'autres Services AWS. Les services suivants sont ceux que vous êtes susceptibles d'utiliser le plus fréquemment :

- Amazon Elastic Compute Cloud (Amazon EC2) — offre une capacité de calcul évolutive dans le AWS Cloud. En utilisant Amazon EC2, vous n'avez pas besoin d'investir dans du matériel au départ, ce qui vous permet de développer et de déployer des applications plus rapidement. Vous pouvez utiliser Amazon EC2 pour lancer autant de serveurs virtuels que nécessaire, configurer la sécurité et les réseaux, et gérer le stockage.
- Amazon Elastic Block Store (Amazon EBS) on Outposts — utilisez les instantanés locaux Amazon EBS sur Outposts pour stocker localement des instantanés de volumes sur un Outpost dans S3 sur Outposts.
- Amazon Relational Database Service (Amazon RDS) on Outposts – utilisez les sauvegardes locales Amazon RDS pour stocker vos sauvegardes Amazon RDS localement sur votre Outpost.
- AWS DataSync— Automatisez le transfert de données entre vos Outposts et Régions AWS choisissez les éléments à transférer, le moment du transfert et la quantité de bande passante réseau à utiliser. S3 on Outposts est intégré à. AWS DataSync Pour les applications locales nécessitant un traitement local à haut débit, S3 sur Outposts fournit un stockage d'objets sur site afin de minimiser les transferts de données et la mémoire tampon liés aux variations réseau, tout en vous offrant la possibilité de transférer facilement des données entre les Outposts et les Régions AWS.

Accès à S3 sur Outposts

Vous pouvez utiliser S3 sur Outposts de l'une des manières suivantes :

AWS Management Console

La console est une interface utilisateur basée sur le Web pour gérer les S3 sur Outposts et les ressources AWS . Si vous vous êtes inscrit à un Compte AWS, vous pouvez accéder à S3 sur

Outposts en vous connectant au AWS Management Console et en choisissant S3 sur la page d' AWS Management Console accueil. Ensuite, sélectionnez Outposts buckets (Compartiments Outpost) dans le panneau de navigation de gauche.

AWS Command Line Interface

Vous pouvez utiliser les outils de ligne de AWS commande pour émettre des commandes ou créer des scripts sur la ligne de commande de votre système afin d'effectuer des tâches AWS (y compris S3).

Le AWS Command Line Interface (AWS CLI) fournit des commandes pour un large éventail de Services AWS. AWS CLI Il est pris en charge sur Windows, macOS et Linux. Consultez le AWS Command Line Interface Guide de l'utilisateur pour démarrer. Pour obtenir plus d'informations sur les commandes que vous pouvez utiliser avec S3 sur Outposts, consultez les sections s3api, s3control, et s3outposts dans la Référence des commandes AWS CLI .

AWS SDK

AWS fournit des SDK (kits de développement logiciel) composés de bibliothèques et d'exemples de code pour différents langages de programmation et plateformes (Java, Python, Ruby, .NET, iOS, Android, etc.). Les AWS SDK constituent un moyen pratique de créer un accès programmatique à S3 sur Outposts et. AWS Parce que S3 sur Outposts utilise les mêmes kits SDK que Amazon S3, S3 sur Outposts offre une expérience cohérente en utilisant les mêmes API, automatisations et outils S3.

S3 sur Outposts est un service REST. Vous pouvez envoyer des requêtes à S3 sur Outposts en utilisant les bibliothèques SDK AWS , qui enveloppent l'API REST sous-jacente et simplifient vos tâches de programmation. Par exemple, ils automatisent les tâches comme le calcul de signatures, la signature cryptographique des demandes, la gestion des erreurs et les nouvelles tentatives automatiques de demande. Pour plus d'informations sur les AWS SDK, notamment sur la façon de les télécharger et de les installer, consultez la section Outils sur AWS lesquels vous pouvez vous appuyer.

Paiement de S3 sur Outposts

Vous pouvez acheter diverses configurations de AWS Outposts rack comprenant une combinaison de types d'instances Amazon EC2, de volumes de disques SSD () à usage général Amazon EBS () et de S3 gp2 on Outposts. Les prix comprennent la livraison, l'installation et la maintenance des services d'infrastructure , ainsi que les correctifs et mises à niveau logiciels.

Pour de plus amples informations, consultez la rubrique Tarification de racks AWS Outposts.

Étapes suivantes

Pour plus d'informations sur l'utilisation de S3 sur Outposts, consultez les rubriques suivantes :

- Configuration de votre Outpost
- En quoi Amazon S3 on Outposts est-il différent de Amazon S3 ?
- Premiers pas avec Amazon S3 sur Outposts
- Mise en réseau pour S3 on Outposts
- Utilisation des compartiments S3 on Outposts
- Utilisation des objets S3 on Outposts
- Sécurité dans S3 on Outposts
- Gestion de stockage S3 on Outposts
- Développement avec Amazon S3 on Outposts

Configuration de votre Outpost

Pour commencer à utiliser Amazon S3 sur Outposts, vous aurez besoin d'un Outpost avec une capacité Amazon S3 déployée sur votre site d'installation. Pour de plus amples informations sur les options de commande d'une capacité Outpost et S3, veuillez consulter AWS Outposts. Pour vérifier si votre Outposts a une capacité S3, vous pouvez utiliser l'appel d'API ListOutpostsWithS3. Pour les spécifications et découvrir en quoi S3 on Outposts est différent d'Amazon S3, veuillez consulter En quoi Amazon S3 on Outposts est-il différent de Amazon S3 ?.

Pour plus d'informations, consultez les rubriques suivantes.

Rubriques

- Commandez un nouvel Outpost

Commandez un nouvel Outpost

Si vous devez commander un nouvel Outpost avec une capacité S3, veuillez consulter Tarification du rack AWS Outposts pour découvrir les options de capacité pour Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS) et Amazon S3.

Après avoir sélectionné votre configuration, suivez les étapes décrites dans Créer un Outpost et commander une capacité Outpost dans le Guide de l'utilisateur AWS Outposts.

En quoi Amazon S3 on Outposts est-il différent de Amazon S3 ?

Amazon S3 sur Outposts fournit du stockage d'objets à votre environnement AWS Outposts sur site. S3 sur Outposts vous aide à répondre aux besoins de traitement local, de résidence des données et de performances exigeantes en maintenant les données à proximité des applications sur site. Grâce aux API et aux fonctions d'Amazon S3 utilisées, S3 sur Outposts facilite le stockage, la sécurisation, le balisage, la création de rapports et le contrôle de l'accès aux données de vos Outposts et étend l'infrastructure AWS à votre installation sur site pour une expérience hybride homogène.

Pour plus d'informations sur le caractère unique de S3 on Outposts, consultez les rubriques suivantes.

Rubriques

- Spécifications de S3 on Outposts
- Opérations API prises en charge par S3 sur Outposts
- Fonctions Simple Storage Service (Amazon S3) non prises en charge par S3 on Outposts
- Exigences réseau de S3 on Outposts

Spécifications de S3 on Outposts

- La taille maximale du compartiment Outposts est de 50 To.
- Le nombre maximal de compartiments Outposts est de 100 par Compte AWS.
- Les compartiments Outposts sont uniquement accessibles à l'aide de points d'accès et de points de terminaison.
- Le nombre maximal de points d'accès par compartiment Outposts est de 10.
- Les stratégies de point d'accès sont limitées à une taille de 20 Ko.
- Le propriétaire d'Outpost peut gérer l'accès au sein de votre organisation dans AWS Organizations à l'aide d'AWS Resource Access Manager. Tous les comptes qui nécessitent un accès à Outpost doivent se trouver au sein de la même organisation que le compte propriétaire dans AWS Organizations.
- Le compte propriétaire du compartiment S3 on Outposts est toujours le propriétaire de tous les objets du compartiment.

- Seul le compte propriétaire du compartiment S3 on Outposts peut effectuer des opérations sur le compartiment.
- Les limitations de taille d'objet sont compatibles avec Simple Storage Service (Amazon S3).
- Tous les objets stockés sur S3 sur Outposts sont stockés dans la classe de stockage `OUTPOSTS`.
- Par défaut, tous les objets stockés dans la classe de stockage `OUTPOSTS` le sont à l'aide du chiffrement côté serveur avec des clés de chiffrement gérées Amazon S3 (SSE-S3). Vous pouvez également choisir explicitement de stocker des objets en utilisant le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C).
- S'il n'y a pas assez d'espace pour stocker un objet sur votre Outpost, l'API renvoie une exception de capacité insuffisante (ICE).

Opérations API prises en charge par S3 sur Outposts

Pour obtenir une liste des opérations API prises en charge par S3 on Outposts, voir Opérations d'API Amazon S3 on Outposts.

Fonctions Simple Storage Service (Amazon S3) non prises en charge par S3 on Outposts

Les fonctions Simple Storage Service (Amazon S3) suivantes ne sont actuellement pas prises en charge par Simple Storage Service (Amazon S3) sur Outposts. Toute tentative de les utiliser est rejetée.

- Listes de contrôle d'accès (ACL)
- Partage des ressources cross-origine (CORS)
- Opérations par lot S3
- Rapports d'inventaire S3
- Modification du chiffrement du compartiment par défaut
- Compartiments publics
- Suppression de l'authentification multifacteur (MFA)
- Transitions de cycle de vie S3 (en plus de la suppression d'objets et de l'arrêt des chargements partitionnés incomplets)
- Mise en suspens juridique du verrouillage des objets S3
- Rétention du verrouillage d'objet

- Chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS)
- Contrôle du délai de réplication S3 (S3 RTC)
- Métriques de demande Amazon CloudWatch
- Configuration des métriques
- Transfer Acceleration
- Notifications d'événements S3
- Compartiments de type Paiement par le demandeur
- S3 Select
- Événements AWS Lambda
- Server access logging (Journalisation des accès au serveur)
- Demandes HTTP POST
- SOAP
- Accès au site web

Exigences réseau de S3 on Outposts

- Pour acheminer les demandes vers un point d'accès S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Les limites suivantes s'appliquent aux points de terminaison pour S3 on Outposts :
  - Chaque cloud privé virtuel (VPC) sur un Outpost peut avoir un point de terminaison associé, et vous pouvez avoir jusqu'à 100 points de terminaison par Outpost.
  - Vous pouvez mapper plusieurs points d'accès au même point de terminaison.
  - Vous ne pouvez ajouter des points de terminaison qu'aux VPC avec des blocs d'adresse CIDR dans les sous-espaces des plages CIDR suivantes :
    - 10.0.0.0/8
    - 172.16.0.0/12
    - 192.168.0.0/16
- Vous ne pouvez créer des points de terminaison vers un Outpost qu'à partir de VPC dont les blocs d'adresse CIDR ne se chevauchent pas.
- Vous ne pouvez créer un point de terminaison qu'à partir de son sous-réseau Outposts.
- Le sous-réseau que vous utilisez pour créer un point de terminaison doit contenir quatre adresses IP que S3 on Outposts peut utiliser.

- Si vous spécifiez le groupe d'adresses IP clients (groupe CoIP), il doit contenir quatre adresses IP que S3 on Outposts peut utiliser.
- Vous ne pouvez créer qu'un point de terminaison par Outpost et par VPC.

Premiers pas avec Amazon S3 sur Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST.

Avec Amazon S3 sur Outposts, vous pouvez utiliser les API et fonctions Amazon S3, telles que le stockage d'objets, les stratégies d'accès, le chiffrement et le balisage, sur AWS Outposts comme vous le faites sur Amazon S3. Pour de plus amples informations sur S3 on Outposts, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Rubriques

- Configuration d'IAM avec S3 on Outposts
- Démarrage à l’aide de la AWS Management Console
- Commencer à utiliser le SDK AWS CLI and pour Java

Configuration d'IAM avec S3 on Outposts

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent les personnes qui peuvent être authentifiées (connectées) et autorisées (disposant d'autorisations) pour utiliser des ressources Amazon S3 on Outposts. IAM est un Service AWS que vous pouvez utiliser sans frais supplémentaires. Par défaut, les utilisateurs IAM ne disposent pas d'autorisations pour des ressources et des opérations S3 on Outposts. Pour accorder des

autorisations d'accès aux ressources et aux opérations d'API de S3 on Outposts, vous pouvez utiliser IAM pour créer des utilisateurs, des groupes ou des rôles et associer des autorisations.

Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :

- Utilisateurs et groupes dans AWS IAM Identity Center :

  Créez un jeu d'autorisations. Suivez les instructions de la rubrique Création d'un jeu d'autorisations du Guide de l'utilisateur AWS IAM Identity Center .
- Utilisateurs gérés dans IAM par un fournisseur d'identité :

  Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique Création d'un rôle pour un fournisseur d'identité tiers (fédération) du Guide de l'utilisateur IAM.
- Utilisateurs IAM :
  - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique Création d'un rôle pour un utilisateur IAM du Guide de l'utilisateur IAM.
  - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique Ajout d'autorisations à un utilisateur (console) du Guide de l'utilisateur IAM.

En plus des politiques basées sur l'identité d'IAM, S3 on Outposts prend en charge les politiques de compartiment et de point d'accès. Les stratégies relatives aux compartiments et aux points d'accès sont des stratégies basées sur les ressources associées à la ressource S3 on Outposts.

- Une stratégie de compartiment est attachée au compartiment et autorise ou refuse les requêtes adressées au compartiment et aux objets qu'il contient en fonction des éléments de la stratégie.
- En revanche, une stratégie de point d'accès est attachée au point d'accès et autorise ou refuse les requêtes adressées au point d'accès.

La stratégie de point d'accès fonctionne avec la stratégie de compartiment associée au compartiment S3 on Outposts. Pour qu'une application ou un utilisateur puisse accéder à des objets dans un compartiment S3 on Outposts via un point d'accès S3 on Outposts, il faut que la politique de point d'accès et la politique de compartiment autorisent la demande.

Les restrictions que vous incluez dans une stratégie de point d'accès s'appliquent uniquement aux demandes effectuées via ce point d'accès. Par exemple, si un point d'accès est attaché à un compartiment, vous ne pouvez pas utiliser la politique de point d'accès pour autoriser ou refuser les

demandes qui sont adressées directement au compartiment. Toutefois, les restrictions que vous imposez à une stratégie de compartiment peuvent autoriser ou refuser les requêtes adressées directement au compartiment ou via le point d'accès.

Dans une politique IAM ou une politique basée sur les ressources, vous définissez quelles actions S3 on Outposts sont autorisées ou refusées. Les actions S3 on Outposts correspondent à des opérations d'API S3 on Outposts spécifiques. Les actions S3 on Outposts utilisent le préfixe de l'espace de noms `s3-outposts:`. Les demandes adressées à l'API de contrôle S3 on Outposts dans un Région AWS et les demandes adressées aux points de terminaison de l'API d'objets sur l'Outpost sont authentifiées à l'aide d'IAM et autorisées par le biais du préfixe d'espace de noms. `s3-outposts:` Pour utiliser S3 on Outposts, configurez vos utilisateurs IAM et autorisez-les par en fonction de l'espace de noms `s3-outposts:`.

Pour plus d'informations, consultez la rubrique Actions, resources, and condition keys for Amazon S3 on Outposts (Actions, ressources et clés de condition pour Amazon S3 on Outposts) dans la section Référence de l'autorisation de service.

> **Note**
>
> - Les listes de contrôle d'accès (ACL) ne sont pas prises en charge par S3 on Outposts.
> - S3 on Outposts considère par défaut le propriétaire du compartiment en tant que propriétaire d'objet, afin de s'assurer que le propriétaire d'un compartiment ne peut pas être empêché d'accéder ou de supprimer des objets.
> - Le blocage de l'accès public S3 est toujours activé pour S3 sur Outposts afin de garantir que les objets ne peuvent jamais avoir un accès public.

Pour plus d'informations sur la configuration d'IAM pour S3 on Outposts, consultez les rubriques suivantes.

Rubriques

- Principes des politiques S3 on Outposts
- Ressources ARN pour S3 on Outposts
- Exemples de stratégies pour S3 on Outposts
- Autorisations pour les points de terminaison S3 on Outposts
- Rôles lié à un service pour S3 sur Outposts

Principes des politiques S3 on Outposts

Lorsque vous créez une stratégie basée sur les ressources pour accorder l'accès à votre compartiment S3 on Outposts, vous devez utiliser l'élément `Principal` pour spécifier la personne ou application qui peut effectuer une requête d'action ou d'opération sur cette ressource. Pour les stratégies S3 on Outposts, vous pouvez utiliser l'un des principes suivants :

- Un Compte AWS
- Un utilisateur IAM
- Un rôle IAM
- Tous les principaux, en utilisant un caractère générique (*) dans une politique qui utilise un élément `Condition` pour limiter l'accès à une plage d'adresses IP spécifique

> **Important**
>
> Vous ne pouvez pas écrire de politique pour un compartiment S3 on Outposts qui utilise un caractère générique (*) dans l'élément `Principal`, sauf si la politique inclut également un élément `Condition` qui restreint l'accès à une plage d'adresses IP spécifique. Cette restriction garantit qu'il n'y a pas d'accès public à votre compartiment S3 on Outposts. Pour obtenir un exemple, consultez Exemples de stratégies pour S3 on Outposts.

Pour en savoir plus sur l'élément `Principal`, consultez AWS JSON policy elements: Principal (Élément de stratégie JSON : Principe) dans le Guide de l'utilisateur IAM.

Ressources ARN pour S3 on Outposts

Les Amazon Resource Names (ARN) pour S3 on Outposts contiennent l'identifiant de l'Outpost en plus de Région AWS l'adresse de l'Outpost, de l'ID et du nom de Compte AWS la ressource. Pour accéder à vos compartiments et à vos objets Outposts et y effectuer des actions, vous devez utiliser l'un des formats ARN présentés dans le tableau suivant.

La *`partition`* valeur de l'ARN fait référence à un groupe de Régions AWS. Chacune Compte AWS est limitée à une partition. Les partitions prises en charge sont les suivantes :

- `aws` – Régions AWS
- `aws-us-gov`— AWS GovCloud (US) Régions

Formats d'ARN pour S3 on Outposts

ARN pour Amazon S3 on Outposts	Format ARN	Exemple
ARN de compartiment	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /bucket/*bucket_name*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /bucket/*example-s3-bucket1*
ARN de point d'accès	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /accesspoint/ *accesspoint_name*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /accesspoint/ *access-point-name*
ARN d'objet	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /bucket/*bucket_name* /object/*object_key*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /bucket/*example-s3-bucket1 /object/myobject*
ARN d'objet de point d'accès S3 onr Outposts (utilisé dans les politiques)	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id* /accesspoint/ *accesspoint_name* /object/*object_key*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904* /accesspoint/ *access-point-name/object/myobject*

ARN pour Amazon S3 on Outposts	Format ARN	Exemple
ARN pour S3 sur Outposts	arn:*partition* :s3-outposts: *region*: *account_id* :outpost/ *outpost_id*	arn:*aws*:s3-outposts: *us-west-2* :*123456789012* :outpost/ *op-01ac5d28a6a232904*

Exemples de stratégies pour S3 on Outposts

Example : politique de compartiment S3 on Outposts avec un principal Compte AWS

La politique de compartiment suivante utilise un Compte AWS principal pour accorder l'accès à un compartiment S3 on Outposts. Pour utiliser cette politique de compartiment, remplacez *user input placeholders* par vos propres informations.

```
{
   "Version":"2012-10-17",
   "Id":"ExampleBucketPolicy1",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
         "Resource":"arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket"
      }
   ]
}
```

Example : politique de compartiment S3 on Outposts avec principal générique (*) et clé de condition pour limiter l'accès à une plage d'adresses IP spécifique

La politique de compartiment suivante utilise un principal générique (*) avec la condition `aws:SourceIp` pour limiter l'accès à une plage d'adresses IP spécifique. Pour utiliser cette politique de compartiment, remplacez *`user input placeholders`* par vos propres informations.

```
{
    "Version": "2012-10-17",
    "Id": "ExampleBucketPolicy2",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": { "AWS" : "*" },
            "Action":"s3-outposts:*",
            "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket",
            "Condition" : {
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "198.51.100.0/24"
                }
            }
        }
    ]
}
```

Autorisations pour les points de terminaison S3 on Outposts

S3 on Outposts nécessite ses propres autorisations dans IAM pour gérer les actions des points de terminaison de S3 on Outposts.

> **Note**
>
> - Pour les points de terminaison qui utilisent le type d'accès du groupe d'adresses IP clients (groupe CoIP), vous devez également disposer des autorisations pour travailler avec des adresses IP à partir de votre groupe CoIP, comme décrit dans le tableau suivant.

- Pour les comptes partagés qui accèdent à S3 sur Outposts en utilisant AWS Resource Access Manager, les utilisateurs de ces comptes partagés ne peuvent pas créer leurs propres points de terminaison sur un sous-réseau partagé. Si un utilisateur d'un compte partagé souhaite gérer ses propres points de terminaison, le compte partagé doit créer son propre sous-réseau sur l'Outpost. Pour plus d'informations, consultez the section called "Partager S3 on Outposts".

Autorisations IAM liées aux points de terminaison S3 sur Outposts

Action	Autorisations IAM
`CreateEndpoint`	`s3-outposts:CreateEndpoint` `ec2:CreateNetworkInterface` `ec2:DescribeNetworkInterfaces` `ec2:DescribeVpcs` `ec2:DescribeSecurityGroups` `ec2:DescribeSubnets` `ec2:CreateTags` `iam:CreateServiceLinkedRole` Pour les points de terminaison qui utilisent le type d'accès du groupe d'adresses IP clients (groupe CoIP) sur site, les autorisations supplémentaires suivantes sont requises : `s3-outposts:CreateEndpoint` `ec2:DescribeCoipPools` `ec2:GetCoipPoolUsage` `ec2:AllocateAddress`

Action	Autorisations IAM
	`ec2:AssociateAddress` `ec2:DescribeAddresses` `ec2:DescribeLocalGatewayRouteTableVpcAssociations`
`DeleteEndpoint`	`s3-outposts:DeleteEndpoint` `ec2:DeleteNetworkInterface` `ec2:DescribeNetworkInterfaces` Pour les points de terminaison qui utilisent le type d'accès du groupe d'adresses IP clients (groupe CoIP) sur site, les autorisations supplémentaires suivantes sont requises : `s3-outposts:DeleteEndpoint` `ec2:DisassociateAddress` `ec2:DescribeAddresses` `ec2:ReleaseAddress`
`ListEndpoints`	`s3-outposts:ListEndpoints`

> **Note**
>
> Vous pouvez utiliser des balises de ressources dans une stratégie IAM pour gérer les autorisations.

Rôles lié à un service pour S3 sur Outposts

S3 sur Outposts utilise des rôles liés à un service IAM pour créer des ressources réseau en votre nom. Pour plus d'informations, voir Utilisation de rôles liés à un service pour Amazon S3 sur Outposts.

Démarrage à l’aide de la AWS Management Console

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour des applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Pour commencer à utiliser S3 on Outposts à l'aide de la console, consultez les rubriques suivantes. Pour commencer à utiliser l’AWS CLI ou AWS SDK for Java, veuillez consulter Commencer à utiliser le SDK AWS CLI and pour Java.

Rubriques

- Créer un compartiment, un point d'accès et un point de terminaison.
- Étapes suivantes

Créer un compartiment, un point d'accès et un point de terminaison.

La procédure suivante vous montre comment créer votre premier compartiment dans S3 on Outposts. Lorsque vous créez un compartiment à l'aide de la console, vous créez également un point d'accès et un point de terminaison associés au compartiment afin que vous puissiez immédiatement commencer à stocker des objets dans votre compartiment.

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Sélectionnez Create Outposts bucket (Créer un compartiment Outposts).
4. Dans Bucket name (Nom du compartiment), saisissez un nom compatible avec le système de nom de domaine (DNS) pour votre compartiment.

   Les nom du compartiment doit présenter les caractéristiques suivantes :

   - Être unique au sein du Compte AWS, de l'Outpost et de la Région AWS où se trouve l'Outpost.
   - Il doit comprendre de 3 à 63 caractères.
   - Ne contient pas de majuscules.
   - Il doit commencer par une minuscule ou un chiffre.

   Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour de plus amples informations sur le choix des noms de compartiment, consultez la section Règles de dénomination de compartiment.

   > **Important**
   >
   > Évitez d'inclure des informations sensibles, notamment des numéros de compte, dans le nom du compartiment. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

5. Pour Outpost, choisissez l'Outpost où vous souhaitez que le compartiment réside.
6. Sous Bucket Versioning (Gestion des versions du compartiment), définissez l'état de gestion des versions S3 pour votre compartiment S3 sur Outposts sur l'une des options suivantes :

   - Disable (Désactiver) (par défaut) : le compartiment reste non versionné.
   - Enable (Activer) : active la gestion des versions S3 pour les objets du compartiment. Tous les objets ajoutés au compartiment reçoivent un ID de version unique.

   Pour de plus amples informations sur la gestion des versions S3, veuillez consulter Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.
7. (Facultatif) Ajoutez les balises facultatives que vous souhaitez associer au compartiment Outposts. Vous pouvez utiliser des balises pour suivre des critères pour des projets individuels

ou des groupes de projets, ou pour étiqueter vos compartiments en utilisant des balises de répartition des coûts.

Par défaut, tous les objets stockés dans votre compartiment Outposts le sont à l'aide du chiffrement côté serveur avec les clés de chiffrement gérées Amazon S3 (SSE-S3). Vous pouvez également choisir explicitement de stocker des objets en utilisant le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C). Pour modifier le type de chiffrement, vous devez utiliser l'API REST, AWS Command Line Interface (AWS CLI), ou les kits SDK AWS.

8. Dans la section Paramètres du point d'accès des Outposts, entrez le nom du point d'accès.

   Les points d'accès S3 on Outpost simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans S3 on Outpost.. Les points d'accès sont des points de terminaison réseau associés à des compartiments Outposts que vous pouvez utiliser pour effectuer des opérations d'objet S3. Pour de plus amples informations, veuillez consulter Points d'accès.

   Les noms des points d'accès doivent être uniques dans le compte pour cette Région et cet Outpost, mais aussi être conformes à Limites et restrictions des points d'accès.

9. Choisissez le VPC pour ce point d'accès Amazon S3 on Outposts.

   Si vous n'avez pas de VPC, choisissez Create VPC (Créer un VPC). Pour de plus amples informations, veuillez consulter Création de points d'accès restreints à un virtual private cloud.

   Un cloud privé virtuel (VPC) vous permet de lancer des ressources AWS dans un réseau virtuel défini par vos soins. Ce réseau virtuel ressemble beaucoup à un réseau traditionnel que vous pourriez exécuter dans votre propre centre de donnés et présente l'avantage d'utiliser l'infrastructure évolutive d AWS.

10. (Opération facultative pour un VPC existant) Choisissez unsous-réseau de point de terminaison pour votre point de terminaison.

    Un sous-réseau est une plage d'adresses IP dans votre VPC. Si vous ne disposez pas du sous-réseau que vous voulez, sélectionnez Create subnet (Créer un sous-réseau). Pour de plus amples informations, veuillez consulter Mise en réseau pour S3 on Outposts.

11. (Opération facultative pour un VPC existant) Choisissez ungroupe de sécurité de point de terminaison pour votre point de terminaison.

    Un groupe de sécurité agit en tant que pare-feu virtuel afin de contrôler le trafic entrant et sortant.

12. (Opération facultative pour un VPC existant) Choisissez leType d'accès au point de terminaison :

- Private (Privé) — à utiliser avec le VPC.
- Customer owned IP (IP appartenant au client) – À utiliser avec un groupe d'adresses IP appartenant au client (groupe CoIP) au sein de votre réseau sur site.

13. (Facultatif) Spécifiez la stratégie de point d'accès à l'Outpost. La console affiche automatiquement le nom Amazon Resource Name (ARN) du point d'accès, que vous pouvez utiliser dans la stratégie.
14. Sélectionnez Create Outposts bucket (Créer un compartiment Outposts).

> **Note**
>
> Cela peut prendre jusqu'à 5 minutes pour que votre point de terminaison Outpost soit créé et que votre compartiment soit prêt à l'emploi. Pour configurer des paramètres de compartiment supplémentaires, sélectionnez View details (Afficher les détails).

Étapes suivantes

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

Après avoir créé un compartiment S3 on Outposts, un point d'accès et un point de terminaison, vous pouvez utiliser AWS CLI ou le kit SDK pour Java pour charger un objet dans votre compartiment. Pour de plus amples informations, veuillez consulter Charger un objet dans un compartiment S3 on Outposts.

Commencer à utiliser le SDK AWS CLI and pour Java

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos AWS Outposts et stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts fournit une nouvelle classe de stockage, S3 Outposts (0UTP0STS), qui utilise les API Amazon S3 et est conçue pour stocker les données de manière durable et redondante sur plusieurs appareils et

serveurs de votre entreprise. AWS Outposts Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts via l'API AWS Management Console, AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Pour démarrer avec S3 on Outposts, vous devez créer un compartiment, un point d'accès et un point de terminaison. Ensuite, vous pouvez charger des objets dans votre compartiment. Les exemples suivants vous montrent comment démarrer avec S3 sur Outposts à l'aide du SDK AWS CLI and pour Java. Pour commencer à utiliser la console, veuillez consulter Démarrage à l'aide de la AWS Management Console.

Rubriques

Étape 1 : Créer un compartiment

Les exemples suivants, AWS CLI ainsi que ceux du SDK pour Java, vous montrent comment créer un bucket S3 on Outposts.

AWS CLI

Example

L'exemple suivant crée un compartiment S3 on Outposts (`s3-outposts:CreateBucket`) à l'aide d' AWS CLI. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

SDK for Java

Example

L'exemple suivant crée un compartiment S3 on Outposts (`s3-outposts:CreateBucket`) à l'aide du kit SDK for Java.

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
            .withBucket(bucketName)
            .withOutpostId(OutpostId)
            .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
 s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();

}
```

Étape 2 : Créer un point d'accès

Pour accéder à votre compartiment Amazon S3 on Outposts, vous devez créer et configurer un point d'accès. Ces exemples vous montrent comment créer un point d'accès à l'aide du SDK AWS CLI et du SDK pour Java.

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès ne prennent en charge que l' virtual-host-style adressage.

AWS CLI

Example

L' AWS CLI exemple suivant crée un point d'accès pour un bucket Outposts. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control create-access-point --account-id 123456789012
 --name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

SDK for Java

Example

L'exemple de kit SDK pour Java suivant illustre la création d'un point d'accès pour un compartiment Outposts. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withName(accessPointName)
            .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
 s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();

}
```

Étape 3 : Créer un point de terminaison

Pour acheminer les demandes vers un point d'accès Amazon S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Pour créer un point de terminaison, vous

devrez disposer d'une connexion active avec votre lien de service vers votre région d'accueil Outpost. Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé. Pour plus d'informations sur les quotas de points de terminaison, consultez Exigences réseau de S3 on Outposts. Vous devez créer un point de terminaison pour pouvoir accéder à vos compartiments Outpost et effectuer des opérations d'objet. Pour plus d'informations, consultez Points de terminaison.

Ces exemples vous montrent comment créer un point de terminaison à l'aide du SDK AWS CLI et du SDK pour Java. Pour de plus amples informations sur les autorisations requises pour créer et gérer des points de terminaison, veuillez consulter Autorisations pour les points de terminaison S3 on Outposts.

AWS CLI

Example

L' AWS CLI exemple suivant crée un point de terminaison pour un Outpost en utilisant le type d'accès aux ressources VPC. Le VPC est dérivé du sous-réseau. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

L' AWS CLI exemple suivant crée un point de terminaison pour un Outpost en utilisant le type d'accès au pool d'adresses IP (pool CoIP) appartenant au client. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

SDK for Java

Example

L'exemple de kit SDK pour Java suivant illustre la création d'un point de terminaison pour un Outpost. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
                .withOutpostId("op-0d79779cef3c30a40")
                .withSubnetId("subnet-8c7a57c5")
                .withSecurityGroupId("sg-ab19e0d1")
                .withAccessType("CustomerOwnedIp")
                .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type
 is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
 s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
 createEndpointResult.getEndpointArn());
}
```

Étape 4 : Charger un objet dans un compartiment S3 on Outposts

Pour charger un objet, consultez Charger un objet dans un compartiment S3 on Outposts

Mise en réseau pour S3 on Outposts

Vous pouvez utiliser Amazon S3 on Outposts pour stocker et récupérer des objets sur site pour les applications qui nécessitent un accès local aux données, un traitement des données et une résidence des données. Cette section décrit les exigences de mise en réseau pour accéder à S3 on Outposts.

Rubriques

- Sélectionner le type d'accès à votre mise en réseau
- Accès à vos compartiments et objets S3 on Outposts
- Interfaces réseau élastiques inter-comptes

Sélectionner le type d'accès à votre mise en réseau

Vous pouvez accéder à S3 sur Outposts à partir d'un VPC ou de votre réseau local. Vous communiquez avec votre compartiment Outpost en utilisant un point d'accès et une connexion de terminaison. Cette connexion maintient le trafic entre votre VPC et vos compartiments S3 on Outposts au sein du réseau AWS. Lorsque vous créez un point de terminaison, vous devez spécifier le type d'accès du point de terminaison, soit `Private` (pour le routage VPC), soit `CustomerOwnedIp` (pour un pool d'adresses IP appartenant au client [pool CoIP]).

- `Private` (pour le routage VPC) — si vous ne spécifiez pas le type d'accès, S3 on Outposts utilise `Private` par défaut. Avec le type d'accès `Private`, les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les ressources de votre Outpost. Vous pouvez travailler avec S3 on Outposts à partir d'un VPC. Ce type de point de terminaison est accessible depuis votre réseau sur site via un routage VPC direct. Pour en savoir plus, consultez Tables de routage de passerelle locale dans le Guide de l'utilisateur AWS Outposts.
- `CustomerOwnedIp` (pour le pool CoIP) — si vous ne choisissez pas le type d'accès `Private` par défaut et sélectionnez `CustomerOwnedIp`, vous devez spécifier une plage d'adresses IP. Vous pouvez utiliser ce type d'accès pour travailler avec S3 on Outposts à partir de votre réseau sur site et au sein d'un VPC. Lorsque vous accédez à S3 sur Outposts dans un VPC, votre trafic est limité à la bande passante de la passerelle locale.

Accès à vos compartiments et objets S3 on Outposts

Pour accéder à vos compartiments et objets S3 sur Outposts, vous devez disposer des éléments suivants :

- Un point d'accès pour le VPC.
- Un point de terminaison pour le même VPC.
- Une connexion active entre votre Outpost et votre Région AWS. Pour en savoir plus sur la connexion de votre Outpost à une Région, consultez Connectivité Outpost vers les Régions AWS dans le guide de l'utilisateur Outposts AWS.

Pour plus d'informations sur l'accès aux compartiments et aux objets dans S3 on Outposts, consultez Utilisation des compartiments S3 on Outposts et Utilisation des objets S3 on Outposts.

Interfaces réseau élastiques inter-comptes

Les points de terminaison S3 on Outposts sont des ressources nommées avec des noms d'Amazon Resource Names (ARN). Lorsque ces points de terminaison sont créés, AWS Outposts met en place quatre interfaces réseau Elastic inter-comptes. Les interfaces réseau Elastic de S3 sur Outposts ressemblent aux autres interfaces réseau à une exception près : S3 sur Outposts associe les interfaces réseau Elastic inter-comptes aux instances Amazon EC2.

Le système de noms de domaine (DNS) de S3 sur Outposts équilibre les charges de vos demandes sur l'interface réseau Elastic inter-comptes. S3 sur Outposts crée l'interface réseau Elastic inter-comptes dans votre compte AWS visible à partir du volet Interfaces réseau de la console Amazon EC2.

Pour les points de terminaison qui utilisent le type d'accès du groupe CoIP, S3 sur Outposts alloue et associe les adresses IP à l'interface réseau Elastic inter-comptes à partir du groupe CoIP configuré.

Utilisation des compartiments S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur Simple Storage Service (Amazon S3), telles que les stratégies d'accès, le chiffrement et le balisage. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Vous communiquez avec vos compartiments Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Pour accéder à vos compartiments et objets S3 on Outposts, vous devez disposer d'un point d'accès pour le VPC et d'un point de terminaison pour le même VPC. Pour de plus amples informations, veuillez consulter Mise en réseau pour S3 on Outposts.

Compartiments

Dans S3 on Outposts, les noms des compartiments sont uniques à un Outpost et nécessitent le code Région AWS de la région où se trouve l'Outpost, l'ID Compte AWS, l'ID de l'Outpost et le nom du compartiment pour les identifier.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

Pour de plus amples informations, veuillez consulter Ressources ARN pour S3 on Outposts.

Points d'accès

Amazon S3 sur Outposts prend en charge les points d'accès Virtual Private Cloud (VPC) uniquement comme seul moyen d'accéder à vos compartiments Outposts.

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

L'exemple suivant montre le format ARN que vous utilisez pour les points d'accès S3 sur Outposts. L'ARN du point d'accès comprend le code Région AWS de la région où se trouve l'Outpost, l'ID Compte AWS, l'ID de l'Outpost et le nom du point d'accès.

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Points de terminaison

Pour acheminer les demandes vers un point d'accès S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Grâce aux points de terminaison S3 sur Outposts, vous pouvez connecter votre VPC en privé à votre compartiment Outpost. Les points de terminaison S3 sur Outposts sont des identificateurs de ressources uniformes (URI) virtuels du point d'entrée de votre compartiment S3 sur Outposts. Il s'agit de composants VPC mis à l'à échelle horizontalement, redondants et hautement disponibles.

Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé, et vous pouvez avoir jusqu'à 100 points de terminaison par Outpost. Vous devez créer ces points de terminaison pour pouvoir accéder à vos compartiments Outpost et effectuer des opérations d'objet. Créer ces points de terminaison permet également au modèle d'API et aux comportements d'être les mêmes en permettant aux mêmes opérations de fonctionner dans S3 et S3 sur Outposts.

Opérations d'API sur S3 on Outposts

Pour gérer les opérations d'API de compartiment Outpost, S3 sur Outposts héberge un point de terminaison distinct du point de terminaison Amazon S3. Ce point de terminaison est `s3-outposts.region.amazonaws.com`.

Pour utiliser les opérations d'API de Amazon S3, vous devez signer le compartiment et les objets en utilisant le format ARN correct. Vous devez transmettre des ARN à l'API afin que Amazon S3 puisse déterminer si la demande concerne Amazon S3 (`s3-control.region.amazonaws.com`) ou S3 on Outposts (`s3-outposts.region.amazonaws.com`). En fonction du format ARN, S3 peut signer et acheminer la demande de manière appropriée.

Chaque fois qu'une demande est envoyée au plan de contrôle Amazon S3, le kit SDK extrait les composants de l'ARN et inclut un en-tête supplémentaire « `x-amz-outpost-id` » avec la valeur du « *outpost-id* » extrait de l'ARN. Le nom de service de l'ARN est utilisé pour signer la demande avant qu'elle ne soit acheminée vers le point de terminaison S3 on Outposts. Ce comportement s'applique à toutes les opérations d'API gérées par le client `s3control`.

Le tableau suivant répertorie les opérations d'API étendues pour Amazon S3 sur Outposts et leurs modifications par rapport à Amazon S3.

API	Valeur de paramètre S3 sur Outposts	
`CreateBucket`	Nom de compartiment en tant qu'ARN, ID Outpost	
`ListRegionalBuckets`	ID Outpost	
`DeleteBucket`	Nom de compartiment en tant qu'ARN	
`DeleteBucketLifecycleConfiguration`	Nom de compartiment en tant qu'ARN	
`GetBucketLifecycleConfiguration`	Nom de compartiment en tant qu'ARN	

API	Valeur de paramètre S3 sur Outposts	
`PutBucketLifecycle Configuration`	Nom de compartiment en tant qu'ARN	
`GetBucketPolicy`	Nom de compartiment en tant qu'ARN	
`PutBucketPolicy`	Nom de compartiment en tant qu'ARN	
`DeleteBucketPolicy`	Nom de compartiment en tant qu'ARN	
`GetBucketTagging`	Nom de compartiment en tant qu'ARN	
`PutBucketTagging`	Nom de compartiment en tant qu'ARN	
`DeleteBucketTagging`	Nom de compartiment en tant qu'ARN	
`CreateAccessPoint`	Nom du point d'accès en tant qu'ARN	
`DeleteAccessPoint`	Nom du point d'accès en tant qu'ARN	
`GetAccessPoint`	Nom du point d'accès en tant qu'ARN	
`GetAccessPoint`	Nom du point d'accès en tant qu'ARN	
`ListAccessPoints`	Nom du point d'accès en tant qu'ARN	

API	Valeur de paramètre S3 sur Outposts	
`PutAccessPointPolicy`	Nom du point d'accès en tant qu'ARN	
`GetAccessPointPolicy`	Nom du point d'accès en tant qu'ARN	
`DeleteAccessPointPolicy`	Nom du point d'accès en tant qu'ARN	

Création et gestion de compartiments S3 on Outposts

Pour plus d'informations sur la création et la gestion des compartiments S3 on Outposts, consultez les rubriques suivantes.

Rubriques

- Création d'un compartiment S3 on Outposts
- Ajout de balises pour les compartiments Amazon S3 on Outposts
- Gestion de l'accès à un compartiment Amazon S3 on Outposts à l'aide d'une stratégie de compartiment
- Lister les compartiments Amazon S3 on Outposts
- Obtenir un compartiment S3 on Outposts en utilisant l’AWS CLI et le kit SDK pour Java
- Suppression de votre compartiment Amazon S3 on Outposts
- Utilisation des points d'accès Amazon S3 on Outposts
- Utilisation des points de terminaison Amazon S3 sur Outposts

Création d'un compartiment S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour des applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur

plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

> **Note**
>
> Le Compte AWS qui crée le compartiment en est le propriétaire et lui peut y valider des actions. Les compartiments possèdent des propriétés de configuration telles que Outpost, balise, chiffrement par défaut et paramètres de point d'accès. Les paramètres du point d'accès comprennent le cloud privé virtuel (VPC), la stratégie du point d'accès pour accéder aux objets du compartiment et d'autres métadonnées. Pour de plus amples informations, veuillez consulter Spécifications de S3 on Outposts.
>
> Si vous souhaitez créer un compartiment qui utilise AWS PrivateLink pour fournir un accès à la gestion des compartiments et des points de terminaison via les points de terminaison d'un VPC d'interface dans votre cloud privé virtuel (VPC), consultez AWS PrivateLink pour S3 on Outposts.

Les exemples suivants vous montrent comment créer un compartiment S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et du kit AWS SDK for Java.

Utilisation de la console S3

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Sélectionnez Create Outposts bucket (Créer un compartiment Outposts).
4. Dans Bucket name (Nom du compartiment), saisissez un nom compatible avec le système de nom de domaine (DNS) pour votre compartiment.

   Les nom du compartiment doit présenter les caractéristiques suivantes :

- Être unique au sein du Compte AWS, de l'Outpost et de la Région AWS où se trouve l'Outpost.
- Il doit comprendre de 3 à 63 caractères.
- Ne contient pas de majuscules.
- Il doit commencer par une minuscule ou un chiffre.

Une fois le compartiment créé, vous ne pouvez pas changer son nom. Pour de plus amples informations sur le choix des noms de compartiment, consultez la section Règles de dénomination de compartiment.

> **Important**
>
> Évitez d'inclure des informations sensibles, notamment des numéros de compte, dans le nom du compartiment. Le nom de compartiment est visible dans les URL qui pointent vers les objets du compartiment.

5. Pour Outpost, choisissez l'Outpost où vous souhaitez que le compartiment réside.
6. Sous Bucket Versioning (Gestion des versions du compartiment), définissez l'état de gestion des versions S3 pour votre compartiment S3 sur Outposts sur l'une des options suivantes :

   - Disable (Désactiver) (par défaut) : le compartiment reste non versionné.
   - Enable (Activer) : active la gestion des versions S3 pour les objets du compartiment. Tous les objets ajoutés au compartiment reçoivent un ID de version unique.

   Pour de plus amples informations sur la gestion des versions S3, veuillez consulter Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.
7. (Facultatif) Ajoutez les balises facultatives que vous souhaitez associer au compartiment Outposts. Vous pouvez utiliser des balises pour suivre des critères pour des projets individuels ou des groupes de projets, ou pour étiqueter vos compartiments en utilisant des balises de répartition des coûts.

   Par défaut, tous les objets stockés dans votre compartiment Outposts le sont à l'aide du chiffrement côté serveur avec les clés de chiffrement gérées Amazon S3 (SSE-S3). Vous pouvez également choisir explicitement de stocker des objets en utilisant le chiffrement côté serveur avec des clés de chiffrement fournies par le client (SSE-C). Pour modifier le type de chiffrement, vous devez utiliser l'API REST, AWS Command Line Interface (AWS CLI), ou les kits SDK AWS.
8. Dans la section Paramètres du point d'accès des Outposts, entrez le nom du point d'accès.

Les points d'accès S3 on Outpost simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans S3 on Outpost.. Les points d'accès sont des points de terminaison réseau associés à des compartiments Outposts que vous pouvez utiliser pour effectuer des opérations d'objet S3. Pour de plus amples informations, veuillez consulter Points d'accès.

Les noms des points d'accès doivent être uniques dans le compte pour cette Région et cet Outpost, mais aussi être conformes à Limites et restrictions des points d'accès.

9. Choisissez le VPC pour ce point d'accès Amazon S3 on Outposts.

   Si vous n'avez pas de VPC, choisissez Create VPC (Créer un VPC). Pour de plus amples informations, veuillez consulter Création de points d'accès restreints à un virtual private cloud.

   Un cloud privé virtuel (VPC) vous permet de lancer des ressources AWS dans un réseau virtuel défini par vos soins. Ce réseau virtuel ressemble beaucoup à un réseau traditionnel que vous pourriez exécuter dans votre propre centre de donnés et présente l'avantage d'utiliser l'infrastructure évolutive d AWS.

10. (Opération facultative pour un VPC existant) Choisissez unsous-réseau de point de terminaison pour votre point de terminaison.

    Un sous-réseau est une plage d'adresses IP dans votre VPC. Si vous ne disposez pas du sous-réseau que vous voulez, sélectionnez Create subnet (Créer un sous-réseau). Pour de plus amples informations, veuillez consulter Mise en réseau pour S3 on Outposts.

11. (Opération facultative pour un VPC existant) Choisissez ungroupe de sécurité de point de terminaison pour votre point de terminaison.

    Un groupe de sécurité agit en tant que pare-feu virtuel afin de contrôler le trafic entrant et sortant.

12. (Opération facultative pour un VPC existant) Choisissez leType d'accès au point de terminaison :

    - Private (Privé) — à utiliser avec le VPC.
    - Customer owned IP (IP appartenant au client) – À utiliser avec un groupe d'adresses IP appartenant au client (groupe CoIP) au sein de votre réseau sur site.

13. (Facultatif) Spécifiez la stratégie de point d'accès à l'Outpost. La console affiche automatiquement le nom Amazon Resource Name (ARN) du point d'accès, que vous pouvez utiliser dans la stratégie.

14. Sélectionnez Create Outposts bucket (Créer un compartiment Outposts).

> **Note**
>
> Cela peut prendre jusqu'à 5 minutes pour que votre point de terminaison Outpost soit créé et que votre compartiment soit prêt à l'emploi. Pour configurer des paramètres de compartiment supplémentaires, sélectionnez View details (Afficher les détails).

Utilisation de AWS CLI

Example

L'exemple suivant crée un compartiment S3 on Outposts (`s3-outposts:CreateBucket`) à l'aide d'AWS CLI. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-
id op-01ac5d28a6a232904
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple suivant crée un compartiment S3 on Outposts (`s3-outposts:CreateBucket`) à l'aide du kit SDK for Java.

```
import com.amazonaws.services.s3control.model.*;

public String createBucket(String bucketName) {

    CreateBucketRequest reqCreateBucket = new CreateBucketRequest()
            .withBucket(bucketName)
            .withOutpostId(OutpostId)
            .withCreateBucketConfiguration(new CreateBucketConfiguration());

    CreateBucketResult respCreateBucket =
 s3ControlClient.createBucket(reqCreateBucket);
    System.out.printf("CreateBucket Response: %s%n", respCreateBucket.toString());

    return respCreateBucket.getBucketArn();
```

```
}
```

Ajout de balises pour les compartiments Amazon S3 on Outposts

Vous pouvez ajouter des balises pour vos compartiments Amazon S3 on Outposts afin de suivre les coûts de stockage ou d'autres critères pour des projets individuels ou des groupes de projets.

> **Note**
>
> Le compte Compte AWS qui crée le compartiment en est le propriétaire, et lui seul peut modifier ses étiquettes.

Utilisation de la console S3

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts dont vous souhaitez modifier les balises.
4. Choisissez l'onglet Propriétés.
5. Sous Balises, choisissez Modifier.
6. Sélectionnez Add new tag (Ajouter une nouvelle balise), puis remplissez le champ Key (Clé) et le champ facultatif Value (Valeur).

   Ajoutez les balises que vous souhaitez associer à un compartiment Outposts afin de suivre d'autres critères pour des projets individuels ou des groupes de projets.
7. Choisissez Enregistrer les modifications.

Utilisation de la AWS CLI

L'exemple AWS CLI suivant applique une configuration de balisage à un compartiment S3 on Outposts en utilisant un document JSON dans le dossier actuel qui spécifie les balises (*`tagging.json`*). Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --tagging file://tagging.json

tagging.json

{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

L'exemple AWS CLI suivant applique une configuration de balisage à un compartiment S3 on Outposts directement depuis la ligne de commande.

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

Pour de plus amples informations sur cette commande, veuillez consulter put-bucket-tagging dans le document AWS CLI Reference.

Gestion de l'accès à un compartiment Amazon S3 on Outposts à l'aide d'une stratégie de compartiment

Une stratégie de compartiment est une stratégie AWS Identity and Access Management (IAM) basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Politique de compartiment.

Vous pouvez mettre à jour votre politique de compartiment pour gérer l'accès à votre compartiment Amazon S3 on Outposts. Pour plus d'informations, consultez les rubriques suivantes.

Rubriques

- Ajout ou modification d'une politique de compartiment pour un compartiment Amazon S3 on Outposts.
- Affichage de la politique de compartiment pour votre compartiment Amazon S3 on Outposts.
- Suppression de la politique de compartiment pour votre compartiment Amazon S3 on Outposts.
- Exemples de stratégie de compartiment

Ajout ou modification d'une politique de compartiment pour un compartiment Amazon S3 on Outposts.

Une stratégie de compartiment est une stratégie AWS Identity and Access Management (IAM) basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Politique de compartiment.

Les rubriques suivantes vous montrent comment mettre à jour votre stratégie de compartiment Amazon S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI), ou du kit AWS SDK for Java.

Utilisation de la console S3

Pour créer ou modifier une stratégie de compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Sélectionnez le compartiment Outpost dont vous souhaitez modifier la politique de compartiment.
4. Choisissez l'onglet Permissions (Autorisations).
5. Dans la section Outposts bucket policy (Politique de compartiment des Outposts), pour créer ou modifier une nouvelle politique, sélectionnez Edit (Modifier).

   Vous pouvez maintenant ajouter ou modifier la stratégie de compartiment S3 on Outposts. Pour plus d'informations, consultez Configuration d'IAM avec S3 on Outposts.

Utilisation de la AWS CLI

L'exemple d'utilisation de la AWS CLI suivant place une stratégie sur un compartiment Outposts.

1. Enregistrez la stratégie de compartiment suivante dans un fichier JSON. Dans cet exemple, le fichier est nommé `policy1.json`. Remplacez *`user input placeholders`* par vos propres informations.

```
{
   "Version":"2012-10-17",
   "Id":"testBucketPolicy",
   "Statement":[
      {
         "Sid":"st1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
         "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket"
      }
   ]
}
```

2. Envoyez le fichier JSON en tant que partie de la commande CLI `put-bucket-policy`. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control put-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --policy file://policy1.json
```

Utilisation du kit AWS SDK pour Java

L'exemple d'utilisation du kit SDK pour Java suivant place une stratégie sur un compartiment Outposts.

```
import com.amazonaws.services.s3control.model.*;
```

```
public void putBucketPolicy(String bucketArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testBucketPolicy\",
\"Statement\":[{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
 AccountId+ "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + bucketArn + "\"}]}";

    PutBucketPolicyRequest reqPutBucketPolicy = new PutBucketPolicyRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withPolicy(policy);

    PutBucketPolicyResult respPutBucketPolicy =
 s3ControlClient.putBucketPolicy(reqPutBucketPolicy);
    System.out.printf("PutBucketPolicy Response: %s%n",
 respPutBucketPolicy.toString());

}
```

Affichage de la politique de compartiment pour votre compartiment Amazon S3 on Outposts.

Une stratégie de compartiment est une stratégie AWS Identity and Access Management (IAM) basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Politique de compartiment.

Les rubriques suivantes vous montrent comment afficher votre stratégie de compartiment Amazon S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI), ou du kit AWS SDK for Java.

Utilisation de la console S3

Pour créer ou modifier une stratégie de compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).

3. Choisissez le compartiment Outposts dont vous souhaitez modifier l'autorisation.
4. Choisissez l'onglet Autorisations.
5. Dans la section Outposts bucket policy (Politique de compartiment des Outposts), vous pouvez passer en revue votre politique de compartiment existante. Pour plus d'informations, consultez Configuration d'IAM avec S3 on Outposts.

Utilisation de la AWS CLI

L'exemple suivant d'utilisation de la AWS CLI obtient une stratégie pour un compartiment Outposts. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Utilisation du kit AWS SDK pour Java

L'exemple de kit SDK pour Java suivant illustre l'obtention d'une stratégie sur un compartiment Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void getBucketPolicy(String bucketArn) {

    GetBucketPolicyRequest reqGetBucketPolicy = new GetBucketPolicyRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    GetBucketPolicyResult respGetBucketPolicy =
 s3ControlClient.getBucketPolicy(reqGetBucketPolicy);
    System.out.printf("GetBucketPolicy Response: %s%n",
 respGetBucketPolicy.toString());

}
```

Suppression de la politique de compartiment pour votre compartiment Amazon S3 on Outposts.

Une stratégie de compartiment est une stratégie AWS Identity and Access Management (IAM) basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Politique de compartiment.

Les rubriques suivantes vous montrent comment afficher votre stratégie de compartiment Amazon S3 on Outposts à l'aide d'AWS Management Console ou d'AWS Command Line Interface (AWS CLI).

Utilisation de la console S3

Pour supprimer une stratégie de compartiment

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts dont vous souhaitez modifier l'autorisation.
4. Choisissez l'onglet Autorisations.
5. Dans la section Outposts bucket policy (Stratégie de compartiment Outposts), sélectionnez Delete (Supprimer).
6. Confirmez la suppression.

Utilisation de la AWS CLI

L'exemple suivant supprime la stratégie de compartiment pour un compartiment S3 on Outposts (`s3-outposts:DeleteBucket`) en utilisant AWS CLI. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

Exemples de stratégie de compartiment

Grâce aux politiques de compartiment S3 on Outposts, vous pouvez sécuriser l'accès aux objets de vos compartiments S3 on Outposts, afin que seuls les utilisateurs disposant des autorisations appropriées puissent y accéder. Vous pouvez même empêcher les utilisateurs authentifiés ne disposant pas des autorisations appropriées d'accéder à vos ressources S3 on Outposts.

Cette section présente des exemples de cas d'utilisation typiques des politiques de compartiment S3 on Outposts. Pour tester ces politiques, remplacez *`user input placeholders`* par vos propres informations (comme le nom de votre compartiment).

Pour accorder ou refuser des autorisations à un ensemble d'objets, vous pouvez utiliser des caractères génériques (*) dans les noms Amazon Resource Name (ARN) et d'autres valeurs. Par exemple, vous pouvez contrôler l'accès aux groupes d'objets qui commencent par un préfixe courant ou se terminent par une extension donnée, comme `.html`.

Pour en savoir plus sur le langage de politique AWS Identity and Access Management (IAM), consultez Configuration d'IAM avec S3 on Outposts.

> **Note**
>
> Lorsque vous testez s3outpostsles autorisations à l'aide de la console Amazon S3, vous devez accorder les autorisations supplémentaires requises par la console `s3outposts:createendpoints3outposts:listendpoints`, telles que,, etc.

Ressources supplémentaires pour créer des politiques de compartiment

- Pour obtenir la liste des actions de politique IAM, des ressources et des clés de condition que vous pouvez utiliser lors de la création d'une politique de bucket S3 on Outposts, consultez Actions, ressources et clés de condition pour Amazon S3 on Outposts.
- Pour obtenir des conseils sur la création de votre politique S3 on Outposts, consultez. Ajout ou modification d'une politique de compartiment pour un compartiment Amazon S3 on Outposts.

Rubriques

- Gestion de l'accès à un bucket Amazon S3 on Outposts en fonction d'adresses IP spécifiques

Gestion de l'accès à un bucket Amazon S3 on Outposts en fonction d'adresses IP spécifiques

Une stratégie de compartiment est une stratégie AWS Identity and Access Management (IAM) basée sur les ressources que vous pouvez utiliser pour accorder des autorisations d'accès à votre compartiment et aux objets qu'il contient. Seul le propriétaire du compartiment peut associer une stratégie à un compartiment. Les autorisations attachées au compartiment s'appliquent à tous les objets du compartiment appartenant au compte propriétaire du compartiment. Les stratégies de compartiment sont limitées à une taille de 20 Ko. Pour plus d'informations, consultez Politique de compartiment.

Restriction de l'accès à des adresses IP spécifiques

L'exemple suivant interdit à tous les utilisateurs d'effectuer des opérations S3 on Outposts sur des objets dans les compartiments spécifiés, sauf si la demande provient de la plage d'adresses IP spécifiée.

> **Note**
>
> Lorsque vous limitez l'accès à une adresse IP spécifique, assurez-vous de spécifier également quels points de terminaison VPC, adresses IP source VPC ou adresses IP externes peuvent accéder au compartiment S3 on Outposts. Dans le cas contraire, vous risquez de perdre l'accès au compartiment si votre politique interdit à tous les utilisateurs d'effectuer s3outpostsdes opérations sur des objets de votre compartiment S3 on Outposts sans que les autorisations appropriées ne soient déjà en place.

La `Condition` déclaration de cette politique identifie *`192.0.2.0/24`*la plage d'adresses IP de version 4 (IPv4) autorisées.

Le bloc `Condition` utilise la condition `NotIpAddress` et la clé de condition `aws:SourceIp`, qui est une clé de condition à l'échelle d'AWS. La clé de condition `aws:SourceIp` ne peut être utilisée que pour les plages d'adresses IP publiques. Pour plus d'informations sur ces clés de condition, consultez Actions, ressources et clés de condition pour S3 on Outposts. Les valeurs IPv4 `aws:SourceIp` font appel à la notation CIDR standard. Pour plus d'informations, consultez la référence aux éléments de politique IAM JSON dans le guide de l'utilisateur IAM.

> **Warning**
>
> Avant d'utiliser cette politique S3 on Outposts, remplacez la plage d'adresses *192.0.2.0/24*IP dans cet exemple par une valeur adaptée à votre cas d'utilisation. Dans le cas contraire, vous ne pourrez plus accéder à votre bucket.

```
{
    "Version": "2012-10-17",
    "Id": "S3OutpostsPolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3outposts:*",
            "Resource": [
                "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
accesspoint/EXAMPLE-ACCESS-POINT-NAME"
                "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET"
            ],
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "192.0.2.0/24"
                }
            }
        }
    ]
}
```

Autoriser les adresses IPv4 et IPv6

Lorsque vous commencez à utiliser des adresses IPv6, nous vous recommandons de mettre à jour toutes les stratégies de votre organisation en y incluant des plages d'adresses IPv6, en plus des plages IPv4 existantes. Cela permettra de s'assurer que les politiques continuent de fonctionner pendant la transition vers IPv6.

L'exemple de politique de compartiment S3 on Outposts suivant montre comment combiner des plages d'adresses IPv4 et IPv6 pour couvrir toutes les adresses IP valides de votre organisation. Dans cet exemple, la politique autorise l'accès aux exemples d'adresses IP

192.0.2.1 et *2001:DB8:1234:5678::1* et le refuse aux adresses *203.0.113.1* et *2001:DB8:1234:5678:ABCD::1*.

La clé de condition `aws:SourceIp` ne peut être utilisée que pour les plages d'adresses IP publiques. Les valeurs IPv6 pour `aws:SourceIp` doivent être au format CIDR standard. Pour IPv6, nous prenons en charge l'utilisation de `::` pour représenter une plage de zéros (par exemple : `2001:DB8:1234:5678::/64`). Pour en savoir plus, consultez Opérateurs de condition d'adresse IP dans le guide de l'utilisateur IAM.

> **Warning**
>
> Remplacez les plages d'adresses IP de cet exemple par des valeurs adaptées à votre cas d'utilisation avant d'utiliser cette politique S3 on Outposts. Dans le cas contraire, vous pourriez perdre la possibilité d'accéder à votre compartiment.

```
{
    "Id": "S3OutpostsPolicyId2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowIPmix",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3outposts:*",
            "Resource": [
                "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET",
                    "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-
ID/bucket/DOC-EXAMPLE-BUCKET/*"
            ],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": [
                        "192.0.2.0/24",
                        "2001:DB8:1234:5678::/64"
                    ]
                },
                "NotIpAddress": {
                    "aws:SourceIp": [
                        "203.0.113.0/24",
```

```
                    "2001:DB8:1234:5678:ABCD::/80"
                ]
            }
        }
    }
]
}
```

Lister les compartiments Amazon S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (OUTPOSTS), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Pour plus d'informations sur l'utilisation des compartiments dans S3 on Outposts, voir Utilisation des compartiments S3 on Outposts.

Les exemples suivants vous montrent comment renvoyer une liste de vos compartiments S3 on Outposts à l'aide de la AWS Management Console, de l'AWS CLI et d'AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Sous Outposts buckets (Compartiments Outposts), vérifiez votre liste de compartiments S3 on Outposts.

Utilisation de la AWS CLI

L'exemple AWS CLI suivant montre l'obtention d'une liste de compartiments dans un Outpost. Pour utiliser cette commande, remplacez chaque *user input placeholder* par vos propres informations. Pour de plus amples informations sur cette commande, consultez list-regional-bucketsdans le document AWS CLI Reference.

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-
id op-01ac5d28a6a232904
```

Utilisation du kit AWS SDK pour Java

L'exemple de kit SDK pour Java suivant illustre l'obtention d'une liste de compartiments dans un Outpost. Pour de plus amples informations, veuillez consulter ListRegionalBuckets dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.services.s3control.model.*;

public void listRegionalBuckets() {

    ListRegionalBucketsRequest reqListBuckets = new ListRegionalBucketsRequest()
            .withAccountId(AccountId)
            .withOutpostId(OutpostId);

    ListRegionalBucketsResult respListBuckets =
 s3ControlClient.listRegionalBuckets(reqListBuckets);
    System.out.printf("ListRegionalBuckets Response: %s%n",
 respListBuckets.toString());

}
```

Obtenir un compartiment S3 on Outposts en utilisant l'AWS CLI et le kit SDK pour Java

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (OUTPOSTS), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment

Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Les exemples suivants vous montrent comment obtenir un compartiment S3 on Outposts à l'aide de l'AWS CLI et d'AWS SDK for Java.

> **Note**
>
> Lorsque vous utilisez Amazon S3 on Outposts via l'AWS CLI ou les kits SDK AWS, vous fournissez l'ARN du point d'accès Outposts à la place du nom du compartiment. L'ARN du point d'accès prend la forme suivante, où *region* est le code Région AWS pour la région où l'Outpost est situé :
> `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point`
> Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Utilisation de la AWS CLI

L'exemple S3 on Outposts suivant obtient un compartiment à l'aide de l'AWS CLI. Pour utiliser cette commande, remplacez chaque *user input placeholder* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter get-bucket dans le document AWS CLI Reference.

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket"
```

Utilisation du kit AWS SDK pour Java

L'exemple S3 on Outposts suivant illustre l'obtention d'un compartiment à l'aide du kit SDK pour Java. Pour de plus amples informations, veuillez consulter GetBucket dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.services.s3control.model.*;

public void getBucket(String bucketArn) {

    GetBucketRequest reqGetBucket = new GetBucketRequest()
            .withBucket(bucketArn)
            .withAccountId(AccountId);

    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);
    System.out.printf("GetBucket Response: %s%n", respGetBucket.toString());

}
```

Suppression de votre compartiment Amazon S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (OUTPOSTS), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Pour plus d'informations sur l'utilisation des compartiments dans S3 on Outposts, voir Utilisation des compartiments S3 on Outposts.

Le compte Compte AWS qui crée le compartiment en est le propriétaire, et lui seul le supprimer.

> **Note**
>
> - Les compartiments Outposts doivent être vides avant de pouvoir être supprimés.

La console Amazon S3 ne prend pas en charge les actions sur les objets S3 on Outposts. Pour supprimer des objets dans un compartiment S3 on Outposts, vous devez utiliser l'API REST, AWS CLI ou les kits SDK AWS.

- Pour pouvoir supprimer un compartiment Outposts, vous devez supprimer tous les points d'accès Outposts pour le compartiment. Pour plus d'informations, consultez Suppression d'un point d'accès.
- Vous ne pouvez pas récupérer un compartiment après l'avoir supprimé.

Les exemples suivants vous montrent comment supprimer un compartiment S3 on Outposts à l'aide d'AWS Management Console et d'AWS Command Line Interface (AWS CLI).

Utilisation de la console S3

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment à supprimer, puis choisissez Delete (Supprimer).
4. Confirmez la suppression.

Utilisation de la AWS CLI

L'exemple suivant supprime un compartiment S3 on Outposts (`s3-outposts:DeleteBucket`) à l'aide d'AWS CLI. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Utilisation des points d'accès Amazon S3 on Outposts

Pour accéder à votre compartiment Amazon S3 on Outposts, vous devez créer et configurer un point d'accès.

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

> **Note**
>
> Le compte Compte AWS qui crée le compartiment Outposts en est le propriétaire, et lui seul peut lui attribuer des points d'accès.

Les sections suivantes décrivent comment créer et gérer des points d'accès pour les compartiments S3 on Outposts.

Rubriques

Création d'un point d'accès S3 on Outposts

Pour accéder à votre compartiment Amazon S3 on Outposts, vous devez créer et configurer un point d'accès.

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points

d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

Les exemples suivants vous montrent comment créer un point d'accès S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et du kit AWS SDK for Java.

> **Note**
>
> Le compte Compte AWS qui crée le compartiment Outposts en est le propriétaire, et lui seul peut lui attribuer des points d'accès.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts pour lequel vous souhaitez créer un point d'accès Outposts.
4. Sélectionnez l'onglet Outposts access points (Points d'accès Outposts).
5. Dans la section Outposts access points (Points d'accès Outposts), choisissez Create Outposts access point (Créer un point d'accès Outposts).
6. Dans Outposts access point settings (Paramètres du point d'accès Outposts), attribuez un nom au point d'accès, puis choisissez le cloud privé virtuel (VPC) du point d'accès.
7. Si vous voulez ajouter une stratégie pour votre point d'accès, saisissez-la dans la section Outposts access point policy (Stratégie de point d'accès Outposts).

   Pour plus d'informations, consultez Configuration d'IAM avec S3 on Outposts.

Utilisation de la AWS CLI

Example

L'exemple AWS CLI suivant crée un point d'accès pour un compartiment Outposts. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control create-access-point --account-id 123456789012
 --name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple de kit SDK pour Java suivant illustre la création d'un point d'accès pour un compartiment Outposts. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.services.s3control.model.*;

public String createAccessPoint(String bucketArn, String accessPointName) {

    CreateAccessPointRequest reqCreateAP = new CreateAccessPointRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withName(accessPointName)
            .withVpcConfiguration(new VpcConfiguration().withVpcId("vpc-12345"));

    CreateAccessPointResult respCreateAP =
 s3ControlClient.createAccessPoint(reqCreateAP);
    System.out.printf("CreateAccessPoint Response: %s%n", respCreateAP.toString());

    return respCreateAP.getAccessPointArn();

}
```

Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts

Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Chaque fois que vous créez un point d'accès pour un compartiment, S3 sur Outposts génère automatiquement un alias de point d'accès. Vous pouvez utiliser cet alias de point d'accès plutôt qu'un ARN de point d'accès pour toutes les opérations de plan de données. Par exemple, vous pouvez utiliser un alias de point d'accès pour effectuer des opérations au niveau de l'objet telles que PUT, GET, LIST, etc. Pour obtenir la liste de ces opérations, veuillez consulter la page Opérations d'API Amazon S3 pour la gestion des objets.

Voici des exemples d'ARN et d'alias de point d'accès pour un point d'accès nommé *my-access-point*.

- ARN du point d'accès – `arn:aws:s3-outposts:`*region*`:`*123456789012*`:outpost/`*op-01ac5d28a6a232904*`/accesspoint/`*my-access-point*
- Alias de point d'accès – *my-access-po*`-`*o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10*`--op-s3`

Pour de plus amples informations sur l'utilisation des ARN, veuillez consulter Amazon Resource Names (ARN) dans le Références générales AWS.

Pour plus d'informations sur les alias de point d'accès, consultez les rubriques suivantes.

Rubriques

- Alias de point d'accès
- Utilisation d'un alias de point d'accès dans une opération d'objet S3 sur Outposts
- Limites

Alias de point d'accès

Un alias de point d'accès est créé dans le même espace de noms qu'un compartiment S3 sur Outposts. Lorsque vous créez un point d'accès, S3 sur Outposts génère automatiquement un alias de point d'accès qui ne peut pas être modifié. Un alias de point d'accès répond à toutes les exigences d'un nom de compartiment S3 sur Outposts valide et comprend les parties suivantes :

access point name prefix`-`*metadata*`--op-s3`

> **Note**
>
> Le suffixe `--op-s3` est réservé aux alias de point d'accès. Nous vous recommandons de ne pas l'utiliser pour les noms de compartiment ou de point d'accès. Pour plus d'informations sur les règles d'attribution de noms des compartiments S3 sur Outposts, consultez Utilisation des compartiments S3 on Outposts.

Recherche de l'alias de point d'accès

Les exemples suivants vous montrent comment trouver un alias de point d'accès à l'aide de la console Amazon S3 et de l'interface AWS CLI.

Example : Rechercher et copier l'alias d'un point d'accès dans la console Amazon S3

Après avoir créé un point d'accès dans la console, vous pouvez obtenir l'alias de point d'accès dans la colonne Access Point alias (Alias de point d'accès) de la liste Access Points (Points d'accès).

Copier l'alias de point d'accès

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Pour copier l'alias de point d'accès, effectuez l'une des opérations suivantes :
   - Dans la liste des Access Points (Points d'accès), sélectionnez le bouton d'option à côté du nom du point d'accès, puis choisissez Copy Access Point alias (Copier un alias de point d'accès).
   - Choisissez le nom du point d'accès. Ensuite, sous Outposts access point overview (Présentation du point d'accès Outposts), copiez l'alias de point d'accès.

Example : Créer un point d'accès en utilisant l'AWS CLI et rechercher l'alias de point d'accès dans la réponse

L'exemple suivant de l'AWS CLI pour la commande `create-access-point` crée le point d'accès et renvoie l'alias de point d'accès généré automatiquement. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control create-access-point --bucket example-outposts-bucket --name example-
outposts-access-point --account-id 123456789012

{
    "AccessPointArn":
    "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
accesspoint/example-outposts-access-point",
    "Alias": "example-outp-o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

Example : Obtenir un alias de point d'accès à l'aide de l'AWS CLI

L'exemple suivant de l'interface AWS CLI pour la commande `get-access-point` récupère les informations relatives au point d'accès spécifié. Ces informations incluent l'alias de point d'accès. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-access-point --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --name example-outposts-access-point --account-id 123456789012

{
    "Name": "example-outposts-access-point",
    "Bucket": "example-outposts-bucket",
    "NetworkOrigin": "Vpc",
    "VpcConfiguration": {
        "VpcId": "vpc-01234567890abcdef"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2022-09-18T17:49:15.584000+00:00",
    "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

Example : Répertorier les points d'accès pour trouver un alias de point d'accès en utilisant l'AWS CLI

L'exemple suivant de l'interface AWS CLI pour la commande `list-access-points` répertorie les informations relatives au point d'accès spécifié. Ces informations incluent l'alias de point d'accès. Pour exécuter cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket

{
    "AccessPointList": [
        {
```

```
            "Name": "example-outposts-access-point",
            "NetworkOrigin": "Vpc",
            "VpcConfiguration": {
                "VpcId": "vpc-01234567890abcdef"
            },
            "Bucket": "example-outposts-bucket",
            "AccessPointArn": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point",
            "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
        }
    ]
}
```

Utilisation d'un alias de point d'accès dans une opération d'objet S3 sur Outposts

Lorsque vous adoptez des points d'accès, vous pouvez utiliser des alias de point d'accès sans nécessiter d'importantes modifications du code.

Cet exemple de l'AWS CLI montre une opération `get-object` pour un compartiment S3 sur Outposts. Cet exemple utilise l'alias de point d'accès comme valeur pour `--bucket` au lieu de l'ARN complet du point d'accès.

```
aws s3api get-object --bucket my-access-po-
o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3 --key  testkey  sample-object.rtf

{
    "AcceptRanges": "bytes",
    "LastModified": "2020-01-08T22:16:28+00:00",
    "ContentLength": 910,
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
    "VersionId": "null",
    "ContentType": "text/rtf",
    "Metadata": {}
}
```

Limites

- Les alias ne peuvent pas être configurés par les clients.
- Les alias ne peuvent pas être supprimés, modifiés ni désactivés sur un point d'accès.

- Vous ne pouvez pas utiliser un alias de point d'accès pour les opérations relatives au plan de contrôle S3 sur Outposts. Pour une liste des opérations du plan de contrôle S3 sur Outposts, consultez Opérations d'API de contrôle Amazon S3 pour la gestion des compartiments.
- Les alias ne peuvent pas être utilisés dans les politiques AWS Identity and Access Management (IAM).

Affichage d'informations sur la configuration d'un point d'accès

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

Les rubriques suivantes vous montrent comment renvoyer les informations de configuration d'un point d'accès S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et d'AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sélectionnez le point d'accès Outpost pour lequel vous souhaitez afficher les détails de la configuration.
4. Sous Outposts access point overview (Aperçu du point d'accès Outpost), passez en revue les détails de la configuration du point d'accès.

Utilisation de la AWS CLI

L'exemple suivant d'utilisation de la AWS CLI obtient un point d'accès pour un compartiment Outposts. Remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Utilisation du kit AWS SDK pour Java

L'exemple de kit SDK pour Java suivant illustre l'obtention d'un point d'accès pour un compartiment Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPoint(String accessPointArn) {

    GetAccessPointRequest reqGetAP = new GetAccessPointRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn);

    GetAccessPointResult respGetAP = s3ControlClient.getAccessPoint(reqGetAP);
    System.out.printf("GetAccessPoint Response: %s%n", respGetAP.toString());

}
```

Afficher une liste de vos points d'accès Amazon S3 on Outposts

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

Les rubriques suivantes vous montrent comment renvoyer une liste de vos points d'accès S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et du kit AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sous Outposts access points (Points d'accès aux Outposts), passez en revue votre liste de points d'accès S3 on Outposts.

Utilisation de la AWS CLI

L'exemple d'utilisation de la AWS CLI suivant répertorie les points d'accès pour un compartiment Outposts. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket
```

Utilisation du kit AWS SDK pour Java

L'exemple de kit SDK pour Java suivant répertorie les points d'accès pour un compartiment Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void listAccessPoints(String bucketArn) {

    ListAccessPointsRequest reqListAPs = new ListAccessPointsRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    ListAccessPointsResult respListAPs = s3ControlClient.listAccessPoints(reqListAPs);
    System.out.printf("ListAccessPoints Response: %s%n", respListAPs.toString());

}
```

Suppression d'un point d'accès

Les points d'accès simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans Amazon S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet Amazon S3, notamment `GetObject` et `PutObject`. Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Les points d'accès prennent uniquement en charge l'adressage de type hôte virtuel.

Les exemples suivants vous montrent comment supprimer un point d'accès à l'aide d'AWS Management Console et d'AWS Command Line Interface (AWS CLI).

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Dans la section Outposts access points (Points d'accès Outposts), choisissez le point d'accès Outposts à supprimer.
4. Sélectionnez Delete.
5. Confirmez la suppression.

Utilisation de la AWS CLI

L'exemple AWS CLI suivant supprime un point d'accès Outposts. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Ajout ou modification d'une stratégie de point d'accès

Les points d'accès disposent d'autorisations et de contrôles de réseau distincts qu'Amazon S3 on Outposts applique pour toute requête effectuée via ce point d'accès. Chaque point d'accès applique une stratégie de point d'accès personnalisée qui fonctionne conjointement avec la stratégie de compartiment associée au compartiment sous-jacent. Pour plus d'informations, consultez Points d'accès.

Les rubriques suivantes vous montrent comment ajouter ou modifier la stratégie de votre point d'accès S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et d'AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts pour lequel vous souhaitez modifier la stratégie de point d'accès.
4. Sélectionnez l'onglet Outposts access points (Points d'accès Outposts).

5. Dans la section Outposts access points (Points d'accès Outposts), sélectionnez le point d'accès dont vous voulez modifier la stratégie, puis Edit policy (Modifier la stratégie).
6. Ajoutez ou modifiez la stratégie dans la section Outposts access point policy (Stratégie de point d'accès Outposts). Pour plus d'informations, consultez Configuration d'IAM avec S3 on Outposts.

Utilisation de la AWS CLI

L'exemple d'utilisation de la AWS CLI suivant place une stratégie sur un point d'accès Outposts.

1. Enregistrez la stratégie de point d'accès suivante dans un fichier JSON. Dans cet exemple, le fichier est nommé `appolicy1.json`. Remplacez *`user input placeholders`* par vos propres informations.

```
{
   "Version":"2012-10-17",
   "Id":"exampleAccessPointPolicy",
   "Statement":[
      {
         "Sid":"st1",
         "Effect":"Allow",
         "Principal":{
            "AWS":"123456789012"
         },
         "Action":"s3-outposts:*",
         "Resource":"arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point
      }
   ]
}
```

2. Envoyez le fichier JSON en tant que partie de la commande CLI `put-access-point-policy`. Remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control put-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --policy file://appolicy1.json
```

Utilisation du kit AWS SDK pour Java

L'exemple suivant d'utilisation du kits SDK pour Java place une stratégie sur un point d'accès Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void putAccessPointPolicy(String accessPointArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testAccessPointPolicy\",
\"Statement\":[{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
 AccountId + "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + accessPointArn +
 "\"}]}";

    PutAccessPointPolicyRequest reqPutAccessPointPolicy = new
 PutAccessPointPolicyRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn)
            .withPolicy(policy);

    PutAccessPointPolicyResult respPutAccessPointPolicy =
 s3ControlClient.putAccessPointPolicy(reqPutAccessPointPolicy);
    System.out.printf("PutAccessPointPolicy Response: %s%n",
 respPutAccessPointPolicy.toString());
    printWriter.printf("PutAccessPointPolicy Response: %s%n",
 respPutAccessPointPolicy.toString());

}
```

Affichage d'une stratégie d'accès pour un point d'accès S3 on Outposts.

Les points d'accès disposent d'autorisations et de contrôles de réseau distincts qu'Amazon S3 on Outposts applique pour toute requête effectuée via ce point d'accès. Chaque point d'accès applique une stratégie de point d'accès personnalisée qui fonctionne conjointement avec la stratégie de compartiment associée au compartiment sous-jacent. Pour plus d'informations, consultez Points d'accès.

Pour plus d'informations sur l'utilisation des points d'accès dans S3 on Outposts, voir Utilisation des compartiments S3 on Outposts.

Les rubriques suivantes vous montrent comment afficher votre stratégie de point d'accès S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et du kit AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sélectionnez le point d'accès Outpost pour lequel vous souhaitez afficher la stratégie.
4. Dans la page Permissions (Autorisations), consultez la stratégie de point d'accès S3 on Outposts.
5. Pour modifier une stratégie de point d'accès, voir Ajout ou modification d'une stratégie de point d'accès.

Utilisation de la AWS CLI

L'exemple d'utilisation de la AWS CLI suivant obtient une stratégie pour un point d'accès Outposts. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Utilisation du kit AWS SDK pour Java

L'exemple d'utilisation du kit SDK pour Java suivant obtient une stratégie pour un point d'accès Outposts.

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPointPolicy(String accessPointArn) {

    GetAccessPointPolicyRequest reqGetAccessPointPolicy = new
 GetAccessPointPolicyRequest()
            .withAccountId(AccountId)
            .withName(accessPointArn);
```

```
    GetAccessPointPolicyResult respGetAccessPointPolicy =
 s3ControlClient.getAccessPointPolicy(reqGetAccessPointPolicy);
    System.out.printf("GetAccessPointPolicy Response: %s%n",
 respGetAccessPointPolicy.toString());
    printWriter.printf("GetAccessPointPolicy Response: %s%n",
 respGetAccessPointPolicy.toString());

}
```

Utilisation des points de terminaison Amazon S3 sur Outposts

Pour acheminer les demandes vers un point d'accès Amazon S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Pour créer un point de terminaison, vous devrez disposer d'une connexion active avec votre lien de service vers votre région d'accueil Outpost. Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé. Pour plus d'informations sur les quotas de points de terminaison, consultez Exigences réseau de S3 on Outposts. Vous devez créer un point de terminaison pour pouvoir accéder à vos compartiments Outpost et effectuer des opérations d'objet. Pour de plus amples informations, veuillez consulter Points de terminaison.

Après avoir créé un point de terminaison, vous pouvez utiliser le champ « Statut » pour comprendre l'état du point de terminaison. Si votre Outpost est hors ligne, il renverra CREATE_FAILED. Vous pouvez vérifier la connexion de votre lien de service, supprimer le point de terminaison et recommencer l'opération de création une fois la connexion rétablie. Vous pouvez consulter une liste des codes d'erreur supplémentaires ci-dessous. Pour de plus amples informations, veuillez consulter Points de terminaison.

API	État	Code d'erreur du motif de l'échec	Message – Motif de l'échec
`CreateEndpoint`	Create_Failed	OutpostNotReachable	Le point de terminaison n'a pas pu être créé car la connexion du lien de service vers votre région d'accueil Outpost est interrompue. Vérifiez votre connexion , supprimez le point de terminaison et réessayez.
`CreateEndpoint`	Create_Failed	InternalError	Le point de terminaison n'a pas pu être créé en raison d'une erreur interne.

API	État	Code d'erreur du motif de l'échec	Message – Motif de l'échec
			Veuillez supprimer le point de terminaison et le créer à nouveau.
`DeleteEndpoint`	Échec de la suppression	OutpostNotReachable	Le point de terminaison n'a pas pu être supprimé car la connexion du lien de service vers votre région d'accueil Outpost est interrompue. Vérifiez votre connexion et réessayez.
`DeleteEndpoint`	Échec de la suppression	InternalError	Le point de terminaison n'a pas pu être supprimé en raison d'une erreur interne. Veuillez réessayer.

Pour de plus amples informations sur l'utilisation des compartiments dans S3 on Outposts, veuillez consulter Utilisation des compartiments S3 on Outposts.

Les sections suivantes décrivent comment créer et gérer des points de terminaison pour S3 on Outposts.

Rubriques

- Création d'un point de terminaison sur un Outpost
- Affichage d'une liste de vos points de terminaison Amazon S3 on Outposts
- Suppression d'un point de terminaison Amazon S3 on Outposts

Création d'un point de terminaison sur un Outpost

Pour acheminer les demandes vers un point d'accès Amazon S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Pour créer un point de terminaison, vous devrez disposer d'une connexion active avec votre lien de service vers votre région d'accueil Outpost. Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé. Pour plus d'informations sur les quotas de points de terminaison, consultez Exigences réseau de S3 on Outposts. Vous devez créer un point de terminaison pour pouvoir accéder à vos compartiments

Outpost et effectuer des opérations d'objet. Pour de plus amples informations, veuillez consulter Points de terminaison.

Autorisations

Pour plus d'informations sur les autorisations requises pour créer un point de terminaison, consultez Autorisations pour les points de terminaison S3 on Outposts.

Lorsque vous créez un point de terminaison, S3 sur Outposts crée également un rôle lié à un service dans votre Compte AWS. Pour de plus amples informations, veuillez consulter Utilisation de rôles liés à un service pour Amazon S3 sur Outposts.

Les exemples suivants vous montrent comment créer un point de terminaison S3 on Outposts à l'aide de la AWS Management Console, de l'AWS Command Line Interface (AWS CLI) et du AWS SDK for Java.

Utilisation de la console S3

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sélectionnez l'onglet Outposts endpoints (Points de terminaison Outposts).
4. Choisissez Create Outposts endpoint (Créer un point de terminaison Outposts).
5. Sous Outpost, sélectionnez l'Outpost sur lequel créer ce point de terminaison.
6. Sous VPC, sélectionnez un VPC qui n'a pas encore de point de terminaison et qui respecte également les règles relatives aux points de terminaison des Outposts.

   Un cloud privé virtuel (VPC) vous permet de lancer des ressources AWS dans un réseau virtuel défini par vos soins. Ce réseau virtuel ressemble beaucoup à un réseau traditionnel que vous pourriez exécuter dans votre propre centre de donnés et présente l'avantage d'utiliser l'infrastructure évolutive d AWS.

   Si vous n'avez pas de VPC, choisissez Create VPC (Créer un VPC). Pour de plus amples informations, veuillez consulter Création de points d'accès restreints à un virtual private cloud.
7. Choisissez Create Outposts endpoint (Créer un point de terminaison Outposts).

Utilisation de AWS CLI

Example

L'exemple AWS CLI suivant crée un point de terminaison pour un Outpost à l'aide du type d'accès aux ressources VPC. Le VPC est dérivé du sous-réseau. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

L'exemple AWS CLI suivant crée un point de terminaison pour un Outpost à l'aide du type d'accès au groupe d'adresses IP clients (groupe CoIP). Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
 subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

Utilisation du kit AWS SDK pour Java

Example

L'exemple de kit SDK pour Java suivant illustre la création d'un point de terminaison pour un Outpost. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.CreateEndpointRequest;
import com.amazonaws.services.s3outposts.model.CreateEndpointResult;

public void createEndpoint() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    CreateEndpointRequest createEndpointRequest = new CreateEndpointRequest()
                .withOutpostId("op-0d79779cef3c30a40")
                .withSubnetId("subnet-8c7a57c5")
                .withSecurityGroupId("sg-ab19e0d1")
                .withAccessType("CustomerOwnedIp")
                .withCustomerOwnedIpv4Pool("ipv4pool-coip-12345678901234567");
```

```
    // Use .withAccessType and .withCustomerOwnedIpv4Pool only when the access type is
    // customer-owned IP address pool (CoIP pool)
    CreateEndpointResult createEndpointResult =
 s3OutpostsClient.createEndpoint(createEndpointRequest);
    System.out.println("Endpoint is created and its ARN is " +
 createEndpointResult.getEndpointArn());
}
```

Affichage d'une liste de vos points de terminaison Amazon S3 on Outposts

Pour acheminer les demandes vers un point d'accès Amazon S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Pour créer un point de terminaison, vous devrez disposer d'une connexion active avec votre lien de service vers votre région d'accueil Outpost. Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé. Pour plus d'informations sur les quotas de points de terminaison, consultez Exigences réseau de S3 on Outposts. Vous devez créer un point de terminaison pour pouvoir accéder à vos compartiments Outpost et effectuer des opérations d'objet. Pour de plus amples informations, veuillez consulter Points de terminaison.

Les exemples suivants vous montrent comment renvoyer une liste ou vos points de terminaison S3 on Outposts à l'aide de la AWS Management Console, de l'AWS Command Line Interface et d'AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sur la page Outposts access points (Points d'accès des Outposts), sélectionnez l'onglet Outposts endpoints (Points de terminaison des Outposts).
4. Sous Outposts endpoints (Points de terminaison Outposts), vous pouvez afficher la liste de vos points de terminaison S3 sur Outposts.

Utilisation de AWS CLI

L'exemple d'utilisation d'AWS CLI suivant répertorie les points de terminaison des ressources AWS Outposts associées à votre compte. Pour de plus amples informations sur cette commande, consultez list-endpoints dans le document AWS CLI Reference.

```
aws s3outposts list-endpoints
```

Utilisation du kit AWS SDK pour Java

L'exemple d'utilisation du kit SDK pour Java suivant génère une liste des points de terminaison pour un Outpost. Pour de plus amples informations, veuillez consulter ListEndpoints dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.ListEndpointsRequest;
import com.amazonaws.services.s3outposts.model.ListEndpointsResult;

public void listEndpoints() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    ListEndpointsRequest listEndpointsRequest = new ListEndpointsRequest();
    ListEndpointsResult listEndpointsResult =
 s3OutpostsClient.listEndpoints(listEndpointsRequest);
    System.out.println("List endpoints result is " + listEndpointsResult);
}
```

Suppression d'un point de terminaison Amazon S3 on Outposts

Pour acheminer les demandes vers un point d'accès Amazon S3 on Outposts, vous devez créer et configurer un point de terminaison S3 on Outposts. Pour créer un point de terminaison, vous devrez disposer d'une connexion active avec votre lien de service vers votre région d'accueil Outpost. Chaque cloud privé virtuel (VPC) sur votre Outpost peut avoir un point de terminaison associé. Pour plus d'informations sur les quotas de points de terminaison, consultez Exigences réseau de S3 on Outposts. Vous devez créer un point de terminaison pour pouvoir accéder à vos compartiments Outpost et effectuer des opérations d'objet. Pour de plus amples informations, veuillez consulter Points de terminaison.

Les exemples suivants vous montrent comment supprimer vos points de terminaison S3 on Outposts à l'aide d'AWS Management Console, d'AWS Command Line Interface (AWS CLI) et du kit AWS SDK for Java.

Utilisation de la console S3

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.

2. Dans le panneau de navigation de gauche, choisissez Outposts access points (Points d'accès Outposts).
3. Sur la page Outposts access points (Points d'accès des Outposts), sélectionnez l'onglet Outposts endpoints (Points de terminaison des Outposts).
4. Sous Outposts endpoints (Points de terminaison des Outposts), sélectionnez le point de terminaison que vous souhaitez supprimer, puis cliquez sur Delete (Supprimer).

Utilisation de AWS CLI

L'exemple d'utilisation de la AWS CLI suivant supprime un point de terminaison pour un Outpost. Pour exécuter cette commande, remplacez *user input placeholders* par vos propres informations.

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-
id op-01ac5d28a6a232904
```

Utilisation du kit AWS SDK pour Java

L'exemple de kit SDK pour Java suivant illustre la suppression d'un point de terminaison pour un Outpost. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.arn.Arn;
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.DeleteEndpointRequest;

public void deleteEndpoint(String endpointArnInput) {
    String outpostId = "op-01ac5d28a6a232904";
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
                .standard().build();

    Arn endpointArn = Arn.fromString(endpointArnInput);
    String[] resourceParts = endpointArn.getResource().getResource().split("/");
    String endpointId = resourceParts[resourceParts.length - 1];
    DeleteEndpointRequest deleteEndpointRequest = new DeleteEndpointRequest()
                .withEndpointId(endpointId)
                .withOutpostId(outpostId);
    s3OutpostsClient.deleteEndpoint(deleteEndpointRequest);
    System.out.println("Endpoint with id " + endpointId + " is deleted.");
```

```
}
```

Utilisation des objets S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos AWS Outposts et stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts fournit une nouvelle classe de stockage, S3 Outposts (OUTPOSTS), qui utilise les API Amazon S3 et est conçue pour stocker les données de manière durable et redondante sur plusieurs appareils et serveurs de votre entreprise. AWS Outposts Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts via l'API AWS Management Console, AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST.

Les objets sont les entités fondamentales stockées dans S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 sur les Outposts, qui inclut le Région AWS code de la région dans laquelle l'Outpost est hébergé, l'ID, l' Compte AWS ID de l'Outpost et le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Les ARN d'objets utilisent le format suivant, qui inclut le lieu d' Région AWS hébergement de l'Outpost, son Compte AWS ID, son identifiant, le nom du bucket et la clé de l'objet :

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/
bucket/example-s3-bucket1/object/myobject
```

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Lorsque vous AWS installez un rack Outpost, vos données restent locales dans votre Outpost afin de répondre aux exigences en matière de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme il AWS Management Console est hébergé dans la région, vous ne pouvez pas utiliser la console pour télécharger ou gérer des objets dans votre Outpost. Cependant, vous pouvez utiliser l'API REST AWS Command Line Interface (AWS CLI) et AWS les SDK pour télécharger et gérer vos objets via vos points d'accès.

Rubriques

Charger un objet dans un compartiment S3 on Outposts

Les objets sont les entités fondamentales stockées dans S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 sur les Outposts, qui inclut le Région AWS code de la région dans laquelle l'Outpost est hébergé, l'ID, l' Compte AWS ID de l'Outpost et le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Lorsque vous AWS installez un rack Outpost, vos données restent locales dans votre Outpost afin de répondre aux exigences en matière de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme il AWS Management Console est hébergé dans la région, vous ne pouvez pas utiliser la console pour télécharger ou gérer des objets dans votre Outpost. Cependant, vous pouvez utiliser l'API REST AWS Command Line Interface (AWS CLI) et AWS les SDK pour télécharger et gérer vos objets via vos points d'accès.

Les AWS SDK for Java exemples suivants AWS CLI vous montrent comment télécharger un objet dans un compartiment S3 on Outposts à l'aide d'un point d'accès.

AWS CLI

Example

L'exemple suivant place un objet nommé `sample-object.xml` dans un compartiment S3 on Outposts (`s3-outposts:PutObject`) à l'aide de l' AWS CLI. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter put-object dans le document AWS CLI Reference.

```
aws s3api put-object --bucket arn:aws:s3-
outposts:Region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --key sample-object.xml --body sample-object.xml
```

SDK for Java

Example

L'exemple suivant place un objet dans un compartiment S3 on Outposts à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Pour plus d'informations, voir Chargement d'objets.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
```

```
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;

public class PutObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Upload a text string as a new object.
            s3Client.putObject(accessPointArn, stringObjKeyName, "Uploaded String
 Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(accessPointArn,
 fileObjKeyName, new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Copie d'un objet dans un compartiment Amazon S3 on Outposts à l'aide du kit AWS SDK for Java

Les objets sont les entités fondamentales stockées dans S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 on Outposts, qui inclut le code Région AWS pour la région où l'Outpost est hébergé, l'ID Compte AWS, l'ID d'Outpost, le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

L'exemple suivant montre comment copier un objet dans un compartiment S3 on Outposts à l'aide du kit AWS SDK for Java.

Utilisation du kit AWS SDK pour Java

L'exemple S3 on Outposts suivant copie un objet dans un nouvel objet situé dans le même compartiment à l'aide du kit SDK for Java. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String sourceKey = "*** Source object key ***";
        String destinationKey = "*** Destination object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(accessPointArn,
 sourceKey, accessPointArn, destinationKey);
            s3Client.copyObject(copyObjectRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Obtenir un objet à partir d'un compartiment Amazon S3 on Outposts

Les objets sont les entités fondamentales stockées dans Amazon S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 on Outposts, qui inclut le code Région AWS pour la région où l'Outpost est hébergé, l'ID Compte AWS, l'ID d'Outpost, le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

Les exemples suivants vous montrent comment télécharger (obtenir) un objet à l'aide de l'AWS Command Line Interface (AWS CLI) et de AWS SDK for Java.

Utilisation de la AWS CLI

L'exemple suivant obtient un objet nommé `sample-object.xml` à partir d'un compartiment S3 on Outposts (`s3-outposts:GetObject`) à l'aide de l'AWS CLI. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter get-object dans le document AWS CLI Reference.

```
aws s3api get-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point --key testkey sample-object.xml
```

Utilisation du kit AWS SDK pour Java

L'exemple S3 on Outposts suivant obtient un objet à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations, veuillez consulter GetObject dans la Référence d'API Amazon Simple Storage Service..

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;

public class GetObject {
    public static void main(String[] args) throws IOException {
        String accessPointArn = "*** access point ARN ***";
        String key = "*** Object key ***";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Get an object and print its contents.
            System.out.println("Downloading an object");
            fullObject = s3Client.getObject(new GetObjectRequest(accessPointArn, key));
            System.out.println("Content-Type: " +
 fullObject.getObjectMetadata().getContentType());
            System.out.println("Content: ");
            displayTextInputStream(fullObject.getObjectContent());

            // Get a range of bytes from an object and print the bytes.
            GetObjectRequest rangeObjectRequest = new GetObjectRequest(accessPointArn,
 key)
                    .withRange(0, 9);
            objectPortion = s3Client.getObject(rangeObjectRequest);
            System.out.println("Printing bytes retrieved.");
            displayTextInputStream(objectPortion.getObjectContent());

            // Get an entire object, overriding the specified response headers, and
 print the object's content.
```

```
            ResponseHeaderOverrides headerOverrides = new ResponseHeaderOverrides()
                    .withCacheControl("No-cache")
                    .withContentDisposition("attachment; filename=example.txt");
            GetObjectRequest getObjectRequestHeaderOverride = new
 GetObjectRequest(accessPointArn, key)
                    .withResponseHeaders(headerOverrides);
            headerOverrideObject = s3Client.getObject(getObjectRequestHeaderOverride);
            displayTextInputStream(headerOverrideObject.getObjectContent());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        } finally {
            // To ensure that the network connection doesn't remain open, close any
 open input streams.
            if (fullObject != null) {
                fullObject.close();
            }
            if (objectPortion != null) {
                objectPortion.close();
            }
            if (headerOverrideObject != null) {
                headerOverrideObject.close();
            }
        }
    }

    private static void displayTextInputStream(InputStream input) throws IOException {
        // Read the text input stream one line at a time and display each line.
        BufferedReader reader = new BufferedReader(new InputStreamReader(input));
        String line = null;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
        System.out.println();
    }
}
```

Liste des objets dans un compartiment Amazon S3 on Outposts

Les objets sont les entités fondamentales stockées dans S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 on Outposts, qui inclut le code Région AWS pour la région où l'Outpost est hébergé, l'ID Compte AWS, l'ID d'Outpost, le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

> **Note**
>
> Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

Les exemples suivants vous montrent comment répertorier les objets d'un compartiment S3 on Outposts à l'aide de l'AWS CLI et d'AWS SDK for Java.

Utilisation de la AWS CLI

L'exemple suivant répertorie les objets dans un compartiment S3 on Outposts (`s3-outposts:ListObjectsV2`) à l'aide de l'AWS CLI. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter list-objects-v2 dans le document AWS CLI Reference.

```
aws s3api list-objects-v2 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

> **Note**
>
> Lorsque vous utilisez cette action avec Amazon S3 sur Outposts via le kit SDK AWS, vous fournissez l'ARN du point d'accès Outposts à la place du nom du compartiment, sous la forme suivante :`arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-Outposts-Access-Point`. Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Utilisation du kit AWS SDK pour Java

L'exemple S3 on Outposts suivant répertorie les objets dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations.

> **Important**
>
> Cet exemple utiliseListObjectsV2, qui est la dernière révision de l'opération d'API `ListObjects`. Nous vous recommandons d'utiliser cette opération d'API révisée pour le développement d'applications. Pour des raisons de rétrocompatibilité, Amazon S3 continue de prendre en charge la version précédente de cette opération d'API.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListObjectsV2 {

    public static void main(String[] args) {
```

```
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            System.out.println("Listing objects");

            // maxKeys is set to 2 to demonstrate the use of
            // ListObjectsV2Result.getNextContinuationToken()
            ListObjectsV2Request req = new
 ListObjectsV2Request().withBucketName(accessPointArn).withMaxKeys(2);
            ListObjectsV2Result result;

            do {
                result = s3Client.listObjectsV2(req);

                for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
                    System.out.printf(" - %s (size: %d)\n", objectSummary.getKey(),
 objectSummary.getSize());
                }
                // If there are more than maxKeys keys in the bucket, get a
 continuation token
                // and list the next objects.
                String token = result.getNextContinuationToken();
                System.out.println("Next Continuation Token: " + token);
                req.setContinuationToken(token);
            } while (result.isTruncated());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Suppression d'objets dans des compartiments Amazon S3 on Outposts

Les objets sont les entités fondamentales stockées dans Amazon S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 on Outposts, qui inclut le code Région AWS pour la région où l'Outpost est hébergé, l'ID Compte AWS, l'ID d'Outpost, le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

Les exemples suivants vous montrent comment supprimer un objet unique ou plusieurs objets dans un compartiment S3 on Outposts à l'aide de l'AWS Command Line Interface (AWS CLI) et de AWS SDK for Java.

Utilisation de la AWS CLI

Les exemples suivants vous montrent comment supprimer un objet unique ou plusieurs objets d'un compartiment S3 on Outposts.

delete-object

L'exemple suivant supprime un objet nommé `sample-object.xml` d'un compartiment S3 on Outposts (`s3-outposts:DeleteObject`) à l'aide de l'AWS CLI. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter delete-object dans le document AWS CLI Reference.

```
aws s3api delete-object --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --key sample-object.xml
```

delete-objects

L'exemple suivant supprime deux objets nommés `sample-object.xml` et `test1.text` d'un compartiment S3 on Outposts (`s3-outposts:DeleteObject`) à l'aide de l'AWS CLI. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour plus d'informations sur cette commande, veuillez consulter delete-objects dans le document AWS CLI Reference.

```
aws s3api delete-objects --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --delete file://delete.json

delete.json
{
  "Objects": [
    {
      "Key": "test1.txt"
    },
    {
      "Key": "sample-object.xml"
    }
  ],
  "Quiet": false
}
```

Utilisation du kit AWS SDK pour Java

Les exemples suivants vous montrent comment supprimer un objet unique ou plusieurs objets d'un compartiment S3 on Outposts.

DeleteObject

L'exemple S3 on Outposts suivant supprime un objet dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, spécifiez l'ARN du point d'accès pour l'Outpost et le nom de clé de l'objet que vous souhaitez supprimer. Pour plus d'informations, veuillez consulter DeleteObject dans la Référence d'API Amazon Simple Storage Service.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** key name ****";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            s3Client.deleteObject(new DeleteObjectRequest(accessPointArn, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

DeleteObjects

L'exemple S3 on Outposts suivant télécharge puis supprime des objets dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, spécifiez l'ARN du point d'accès pour l'Outpost. Pour de plus amples informations, veuillez consulter DeleteObject dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;

import java.util.ArrayList;

public class DeleteObjects {

    public static void main(String[] args) {
       String accessPointArn = "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point";


        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Upload three sample objects.
            ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
            for (int i = 0; i < 3; i++) {
                String keyName = "delete object example " + i;
                s3Client.putObject(accessPointArn, keyName, "Object number " + i + "
 to be deleted.");
                keys.add(new KeyVersion(keyName));
            }
            System.out.println(keys.size() + " objects successfully created.");
```

```
            // Delete the sample objects.
            DeleteObjectsRequest multiObjectDeleteRequest = new
 DeleteObjectsRequest(accessPointArn)
                    .withKeys(keys)
                    .withQuiet(false);

            // Verify that the objects were deleted successfully.
            DeleteObjectsResult delObjRes =
 s3Client.deleteObjects(multiObjectDeleteRequest);
            int successfulDeletes = delObjRes.getDeletedObjects().size();
            System.out.println(successfulDeletes + " objects successfully
 deleted.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Utilisation de HeadBucket pour déterminer si un compartiment S3 on Outposts existe et que vous disposez des autorisations d'accès

Les objets sont les entités fondamentales stockées dans Amazon S3 on Outposts. Chaque objet est contenu dans un compartiment. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous spécifiez le compartiment pour les opérations d'objet, vous utilisez l'Amazon Resource Name (ARN) du point d'accès ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

L'exemple suivant montre le format ARN pour les points d'accès S3 on Outposts, qui inclut le code Région AWS pour la région où l'Outpost est hébergé, l'ID Compte AWS, l'ID d'Outpost, le nom du point d'accès :

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

Pour plus d'informations sur les ARN de S3 on Outposts, consultez Ressources ARN pour S3 on Outposts.

> **Note**
>
> Avec Amazon S3 on Outposts, les données des objets sont toujours stockées sur l'Outpost. Quand AWS installe un rack Outpost, vos données restent locales à votre Outpost pour répondre aux exigences de résidence des données. Vos objets ne quittent jamais votre Outpost et ne se trouvent pas dans un Région AWS. Comme AWS Management Console est hébergé dans la région, vous ne pouvez pas l'utiliser pour charger ou gérer des objets dans votre Outpost. Toutefois, vous pouvez utiliser l'API REST, l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS pour charger et gérer vos objets via vos points d'accès.

Les exemples AWS Command Line Interface (AWS CLI) et AWS SDK for Java suivants vous montrent comment utiliser l'opération d'API HeadBucket pour déterminer si un compartiment Amazon S3 on Outposts existe et si vous avez l'autorisation d'y accéder. Pour de plus amples informations, veuillez consulter HeadBucket dans le document Amazon Simple Storage Service API Reference.

Utilisation de la AWS CLI

L'exemple AWS CLI suivant de S3 on Outposts utilise la commande `head-bucket` pour déterminer si un compartiment existe et si vous avez les autorisations pour y accéder. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, veuillez consulter head-bucket dans le document AWS CLI Reference.

```
aws s3api head-bucket --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

Utilisation du kit AWS SDK pour Java

L'exemple suivant de S3 on Outposts montre comment déterminer si un compartiment existe et si vous avez l'autorisation d'y accéder. Pour utiliser cet exemple, spécifiez l'ARN du point d'accès pour l'Outpost. Pour de plus amples informations, veuillez consulter HeadBucket dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.HeadBucketRequest;

public class HeadBucket {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            s3Client.headBucket(new HeadBucketRequest(accessPointArn));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Réalisation et gestion d'un chargement partitionné avec le kit SDK for Java.

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos ressources AWS Outposts afin de stocker et récupérer des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Les exemples suivants montrent comment vous pouvez utiliser S3 on Outposts avec AWS SDK for Java pour lancer et gérer un chargement partitionné.

Rubriques

- Effectuer le chargement partitionné d'un objet dans un compartiment S3 sur Outposts
- Copie d'un objet de grande taille dans un compartiment S3 on Outposts à l'aide du chargement partitionné
- Générer une liste des parties d'un objet dans un compartiment S3 on Outposts
- Récupérer une liste de chargements partitionnés en cours dans un compartiment S3 on Outposts

Effectuer le chargement partitionné d'un objet dans un compartiment S3 sur Outposts

L'exemple S3 on Outposts suivant lance, télécharge et achève le chargement partitionné d'un objet dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Pour plus d'informations, consultez Chargement d'un objet à l'aide du chargement partitionné.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
```

```
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(accessPointArn)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(accessPointArn)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }

            // Complete the upload request to concatenate all uploaded parts and make
the copied object available.
            CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
                    accessPointArn,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
```

Copie d’un objet de grande taille dans un compartiment S3 on Outposts à l'aide du chargement partitionné

L'exemple S3 on Outposts suivant copie un objet dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Cet exemple est adapté de Copie d'un objet à l'aide du chargement partitionné.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
```

```
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
 InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
 s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
 s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(accessPointArn)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(accessPointArn)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }
```

```
            // Complete the upload request to concatenate all uploaded parts and make
 the copied object available.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest(
                    accessPointArn,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

Générer une liste des parties d'un objet dans un compartiment S3 on Outposts

L'exemple S3 on Outposts suivant répertorie les parties d'un objet dans un compartiment à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;
```

```
import java.util.List;

public class ListParts {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** Key name ***";
        String uploadId = "*** Upload ID ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            ListPartsRequest listPartsRequest = new ListPartsRequest(accessPointArn,
 keyName, uploadId);
            PartListing partListing = s3Client.listParts(listPartsRequest);
            List<PartSummary> partSummaries = partListing.getParts();

            System.out.println(partSummaries.size() + " multipart upload parts");
            for (PartSummary p : partSummaries) {
                System.out.println("Upload part: Part number = \"" + p.getPartNumber()
 + "\", ETag = " + p.getETag());
            }

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Récupérer une liste de chargements partitionnés en cours dans un compartiment S3 on Outposts

L'exemple S3 sur Outposts suivant montre comment récupérer une liste de chargements partitionnés en cours à partir d'un compartiment Outposts à l'aide du kit SDK pour Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Il s'agit d'un exemple adapté de l'exemple Liste des chargements partitionnés pour Amazon S3.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

import java.util.List;

public class ListMultipartUploads {
    public static void main(String[] args) {
                String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .enableUseArnRegion()
                    .build();

            // Retrieve a list of all in-progress multipart uploads.
            ListMultipartUploadsRequest allMultipartUploadsRequest = new
 ListMultipartUploadsRequest(accessPointArn);
            MultipartUploadListing multipartUploadListing =
 s3Client.listMultipartUploads(allMultipartUploadsRequest);
            List<MultipartUpload> uploads =
 multipartUploadListing.getMultipartUploads();

            // Display information about all in-progress multipart uploads.
            System.out.println(uploads.size() + " multipart upload(s) in progress.");
            for (MultipartUpload u : uploads) {
                System.out.println("Upload in progress: Key = \"" + u.getKey() + "\",
 id = " + u.getUploadId());
```

```
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Utilisation d'URL présignée pour S3 on Outposts

Pour accorder un accès limité dans le temps aux objets stockés localement dans un Outpost sans mettre à jour votre stratégie de compartiment, vous pouvez utiliser une URL présignée. Avec les URL présignées, vous pouvez, en tant que propriétaire du compartiment, partager des objets avec des personnes dans votre cloud privé virtuel (VPC) ou leur accorder la possibilité de télécharger ou de supprimer des objets.

Lorsque vous créez une URL présignée à l'aide de kits SDK AWS ou de AWS Command Line Interface (AWS CLI), vous associez l'URL à une action spécifique. Vous accordez également un accès limité dans le temps à l'URL présignée en choisissant un délai d'expiration personnalisé qui peut aller de 1 seconde à 7 jours. Lorsque vous partagez l'URL présignée, la personne dans le VPC peut effectuer l'action intégrée dans l'URL comme s'il s'agissait de l'utilisateur connecté d'origine. Lorsque l'URL atteint son délai d'expiration, elle expire et ne fonctionne plus.

Limitation des capacités des URL présignées

Les capacités d'une URL présignée sont limitées par les autorisations de l'utilisateur qui l'a créée. En résumé, les URL présignées correspondent à des jetons porteurs qui donnent accès à ceux qui les possèdent. À ce titre, nous vous recommandons de les protéger de manière appropriée.

AWS Signature Version 4 (SigV4)

Pour imposer un comportement spécifique lorsque les requêtes d'URL présignées sont authentifiées à l'aide d'AWS Signature Version 4 (SigV4), vous pouvez utiliser les clés de condition dans les stratégies de compartiment et les stratégies de point d'accès. Par exemple, vous pouvez créer une stratégie de compartiment qui utilise la condition `s3-outposts:signatureAge` pour refuser toute

demande d'URL présignée Amazon S3 on Outposts sur les objets du compartiment `example-outpost-bucket` si la signature date de plus de 10 minutes. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Deny a presigned URL request if the signature is more than 10
 minutes old",
            "Effect": "Deny",
            "Principal": {"AWS":"444455556666"},
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
            "Condition": {
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
            }
        }
    ]
}
```

Pour obtenir la liste des clés de condition et des exemples de stratégies supplémentaires que vous pouvez utiliser pour imposer un comportement spécifique lorsque des requêtes d'URL présignées sont authentifiées à l'aide de Signature Version 4, consultez Clés de stratégie spécifiques à l'authentification AWS Signature Version 4 (SigV4).

Restriction de chemin réseau

Si vous souhaitez restreindre l'utilisation des URL présignées et de tous les accès S3 on Outposts à des chemins réseau particuliers, vous pouvez écrire des stratégies qui nécessitent un chemin d'accès réseau particulier. Pour définir la restriction sur le principal IAM qui effectue l'appel, vous pouvez utiliser des politiques IAM AWS Identity and Access Management (par exemple, utilisateur, groupe ou stratégies de rôle). Pour définir la restriction de la ressource S3 on Outposts, vous pouvez utiliser des stratégies basées sur les ressources (par exemple, des stratégies de compartiment et de point d'accès).

Une restriction de chemin réseau sur le principal IAM exige que l'utilisateur de ces informations d'identification effectue des requêtes à partir du réseau spécifié. Une restriction sur le compartiment

ou le point d'accès nécessite que toutes les requêtes adressées à cette ressource proviennent du réseau spécifié. Ces restrictions s'appliquent également hors du scénario des URL présignées.

La condition globale IAM que vous utilisez dépend du type de point de terminaison. Si vous utilisez le point de terminaison public pour S3 on Outposts, utilisez `aws:SourceIp`. Si vous utilisez le point de terminaison d'un VPC pour S3 sur Outposts, utilisez `aws:SourceVpc` ou `aws:SourceVpce`.

La déclaration de politique IAM suivante nécessite que le principal accède à AWS uniquement à partir de la plage réseau spécifiée. Avec cette déclaration de stratégie, tous les accès doivent provenir de cette plage, même lorsqu'une personne utilise une URL présignée pour S3 on Outposts. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Pour un exemple de stratégie de compartiment utilisant la clé de condition globale `aws:SourceIP` AWS pour restreindre l'accès à un compartiment S3 on Outposts à une plage réseau spécifique, consultez Configuration d'IAM avec S3 on Outposts.

Utilisateurs habilités à créer une URL présignée

Toute personne qui possède des autorisations de sécurité valides peut créer une URL présignée. Mais pour qu'un utilisateur dans le VPC puisse accéder correctement à un objet, l'URL présignée doit être créée par une personne qui possède l'autorisation d'effectuer l'opération sur laquelle l'URL présignée est basée.

Vous pouvez utiliser les informations d'identification suivantes pour créer une URL présignée :

- Profil d'instance IAM : valide pendant 6 heures.
- AWS Security Token Service : valide pendant 36 heures lorsque signé avec des autorisations permanentes, telles que les autorisations de l'utilisateur root du Compte AWS ou d'un utilisateur IAM.

- Utilisateur IAM : valide pendant 7 jours en cas d'utilisation d'AWS Signature Version 4.

  Afin de créer une URL présignée valide pendant 7 jours, commencez par déléguer des autorisations d'utilisateur IAM (clé d'accès et clé secrète) pour le kit SDK que vous utilisez. Ensuite, générez une URL présignée en utilisant AWS Signature Version 4.

> **Note**
>
> - Si vous avez créé une URL présignée à l'aide d'un jeton temporaire, l'URL expire lorsque le jeton expire, même si vous avez créé l'URL avec une heure d'expiration postérieure.
> - Étant donné que les URL présignées accordent l'accès à vos compartiments S3 on Outposts à toute personne possédant l'URL, nous vous recommandons de les protéger de manière appropriée. Pour en savoir plus sur la protection des URL présignées, veuillez consulter Limitation des capacités des URL présignées.

Quand S3 on Outposts vérifie-t-il la date et l'heure d'expiration dans une URL présignée ?

S3 on Outposts vérifie la date et l'heure d'expiration d'une URL signée au moment de la requête HTTP. Par exemple, si un client commence à télécharger un fichier volumineux immédiatement avant la date d'expiration, le téléchargement continue même si la date d'expiration intervient pendant le téléchargement. Cependant, si la connexion est perdue et que le client essaie de redémarrer le téléchargement une fois la date d'expiration passée, le téléchargement échoue.

Pour plus d'informations sur l'utilisation d'une URL présignée pour partager ou charger des objets, consultez les rubriques suivantes.

Rubriques

- Partage d'objets à l'aide d'URL présignées
- Génération d'une URL présignée pour charger un objet sur un compartiment S3 on Outposts

Partage d'objets à l'aide d'URL présignées

Pour accorder un accès limité dans le temps aux objets stockés localement dans un Outpost sans mettre à jour votre stratégie de compartiment, vous pouvez utiliser une URL présignée. Avec les URL

présignées, vous pouvez, en tant que propriétaire du compartiment, partager des objets avec des personnes dans votre cloud privé virtuel (VPC) ou leur accorder la possibilité de télécharger ou de supprimer des objets.

Lorsque vous créez une URL présignée à l'aide AWS des SDK ou du AWS Command Line Interface (AWS CLI), vous associez l'URL à une action spécifique. Vous accordez également un accès limité dans le temps à l'URL présignée en choisissant un délai d'expiration personnalisé qui peut aller de 1 seconde à 7 jours. Lorsque vous partagez l'URL présignée, la personne dans le VPC peut effectuer l'action intégrée dans l'URL comme s'il s'agissait de l'utilisateur connecté d'origine. Lorsque l'URL atteint son délai d'expiration, elle expire et ne fonctionne plus.

Lorsque vous créez une URL présignée, vous devez fournir vos informations d'identification de sécurité, puis spécifier les éléments suivants :

- Un point d'accès pour Amazon Resource Name (ARN) du compartiment S3 on Outposts
- Une clé d'objet
- Une méthode HTTP (`GET` pour télécharger des objets)
- Une date et une heure d'expiration

Une URL présignée est uniquement valide pendant la durée spécifiée. Autrement dit, vous devez commencer l'action autorisée par l'URL avant la date et l'heure d'expiration. Vous pouvez utiliser une URL présignée plusieurs fois, jusqu'à la date et l'heure d'expiration. Si vous avez créé une URL présignée à l'aide d'un jeton temporaire, alors l'URL expire lorsque le jeton expire, même si vous avez créé l'URL avec une heure d'expiration postérieure.

Les utilisateurs du cloud privé virtuel (VPC) qui ont accès à l'URL présignée peuvent accéder à l'objet. Par exemple, si votre compartiment contient une vidéo et que ce compartiment et l'objet sont confidentiels, vous pouvez partager la vidéo avec d'autres en générant une URL présignée. Étant donné que les URL présignées accordent l'accès à vos compartiments S3 on Outposts à toute personne possédant l'URL, nous vous recommandons de protéger ces URL de manière appropriée. Pour plus d'informations sur la protection des URL présignées, veuillez consulter la section Limitation des capacités des URL présignées.

Toute personne qui possède des autorisations de sécurité valides peut créer une URL présignée. Cependant, l'URL présignée doit être créée par une personne disposant des autorisations nécessaires pour effectuer l'opération sur laquelle l'URL présignée est basée. Pour plus d'informations, consultez Utilisateurs habilités à créer une URL présignée.

Vous pouvez générer une URL présignée pour partager un objet dans un compartiment S3 on Outposts à l'aide des kits SDK AWS et de la AWS CLI. Pour plus d'informations, consultez les exemples suivants.

Utilisation des AWS SDK

Vous pouvez utiliser les AWS SDK pour générer une URL présignée que vous pouvez communiquer à d'autres personnes afin qu'elles puissent récupérer un objet.

> **Note**
>
> Lorsque vous utilisez les AWS SDK pour générer une URL présignée, le délai d'expiration maximal d'une URL présignée est de 7 jours à compter de sa création.

Java

Example

L'exemple suivant génère une URL présignée que vous pouvez communiquer à d'autres afin qu'ils puissent récupérer un objet depuis un compartiment S3 on Outposts. Pour plus d'informations, consultez Utilisation d'URL présignée pour S3 on Outposts. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

Pour obtenir des instructions sur la création et le test d'un échantillon fonctionnel, voir Getting Started dans le guide du AWS SDK for Java développeur.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {
```

```
    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accessPointArn = "*** access point ARN ***";
        String objectKey = "*** object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Set the presigned URL to expire after one hour.
            java.util.Date expiration = new java.util.Date();
            long expTimeMillis = Instant.now().toEpochMilli();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);

            // Generate the presigned URL.
            System.out.println("Generating pre-signed URL.");
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                    new GeneratePresignedUrlRequest(accessPointArn, objectKey)
                            .withMethod(HttpMethod.GET)
                            .withExpiration(expiration);
            URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);

            System.out.println("Pre-Signed URL: " + url.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't
 process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Example

L'exemple suivant génère une URL présignée que vous pouvez communiquer à d'autres afin qu'ils puissent récupérer un objet depuis un compartiment S3 on Outposts. Pour plus d'informations, consultez Utilisation d'URL présignée pour S3 on Outposts. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

Pour plus d'informations sur la configuration et l'exécution des exemples de code, consultez Getting Started with the AWS SDK for .NET dans AWS le Guide du développeur du SDK pour .NET.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;

namespace Amazon.DocSamples.S3
{
    class GenPresignedURLTest
    {
        private const string accessPointArn = "*** access point ARN ***";
        private const string objectKey = "*** object key ***";
        // Specify how long the presigned URL lasts, in hours.
        private const double timeoutDuration = 12;
        // Specify your bucket Region (an example Region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            string urlString = GeneratePreSignedURL(timeoutDuration);
        }
        static string GeneratePreSignedURL(double duration)
        {
            string urlString = "";
            try
            {
                GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
                {
```

```
                    BucketName = accessPointArn,
                    Key = objectKey,
                    Expires = DateTime.UtcNow.AddHours(duration)
                };
                urlString = s3Client.GetPreSignedURL(request1);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            return urlString;
        }
    }
}
```

Python

L'exemple suivant génère une URL présignée pour partager un objet à l'aide du SDK pour Python (Boto3). Par exemple, utilisez un client Boto3 et la fonction `generate_presigned_url` pour générer une URL présignée qui vous permet d'accéder à un objet `GET`.

```
import boto3
    url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

Pour plus d'informations sur l'utilisation du kit SDK pour Python (Boto3) pour générer une URL présignée, consultez la section Python dans la Référence d'API AWS SDK for Python (Boto) .

En utilisant le AWS CLI

L'exemple de AWS CLI commande suivant génère une URL présignée pour un compartiment S3 on Outposts. Pour utiliser cet exemple, remplacez *`user input placeholders`* par vos propres informations.

> **Note**
>
> Lorsque vous utilisez le AWS CLI pour générer une URL présignée, le délai d'expiration maximal d'une URL présignée est de 7 jours à compter de sa création.

```
aws s3 presign s3://arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point/mydoc.txt --expires-in 604800
```

Pour de plus amples informations, veuillez consulter presign (présigner) dans la Référence des commandes AWS CLI .

Génération d'une URL présignée pour charger un objet sur un compartiment S3 on Outposts

Pour accorder un accès limité dans le temps aux objets stockés localement dans un Outpost sans mettre à jour votre stratégie de compartiment, vous pouvez utiliser une URL présignée. Avec les URL présignées, vous pouvez, en tant que propriétaire du compartiment, partager des objets avec des personnes dans votre cloud privé virtuel (VPC) ou leur accorder la possibilité de télécharger ou de supprimer des objets.

Lorsque vous créez une URL présignée à l'aide de kits SDK AWS ou de AWS Command Line Interface (AWS CLI), vous associez l'URL à une action spécifique. Vous accordez également un accès limité dans le temps à l'URL présignée en choisissant un délai d'expiration personnalisé qui peut aller de 1 seconde à 7 jours. Lorsque vous partagez l'URL présignée, la personne dans le VPC peut effectuer l'action intégrée dans l'URL comme s'il s'agissait de l'utilisateur connecté d'origine. Lorsque l'URL atteint son délai d'expiration, elle expire et ne fonctionne plus.

Lorsque vous créez une URL présignée, vous devez fournir vos informations d'identification de sécurité, puis spécifier les éléments suivants :

- Un point d'accès pour Amazon Resource Name (ARN) du compartiment S3 on Outposts
- Une clé d'objet
- Une méthode HTTP (PUT pour le chargement d'objets)
- Une date et une heure d'expiration

Une URL présignée est uniquement valide pendant la durée spécifiée. Autrement dit, vous devez commencer l'action autorisée par l'URL avant la date et l'heure d'expiration. Vous pouvez utiliser une URL présignée plusieurs fois, jusqu'à la date et l'heure d'expiration. Si vous avez créé une URL présignée à l'aide d'un jeton temporaire, alors l'URL expire lorsque le jeton expire, même si vous avez créé l'URL avec une heure d'expiration postérieure.

Si l'action autorisée par l'URL présignée est composée de plusieurs étapes, comme un chargement partitionné, vous devez démarrer l'ensemble des étapes avant l'expiration. Si S3 on Outposts tente de démarrer une étape avec une URL expirée, vous recevrez une erreur.

Les utilisateurs du cloud privé virtuel (VPC) qui ont accès à l'URL présignée peuvent charger des objets. Par exemple, un utilisateur du VPC qui a accès à l'URL présignée peut charger un objet dans votre compartiment. Étant donné que les URL présignées accordent l'accès à votre compartiment S3 on Outposts à tout utilisateur possédant l'URL présignée, nous vous recommandons de protéger ces URL de manière appropriée. Pour plus d'informations sur la protection des URL présignées, veuillez consulter la section Limitation des capacités des URL présignées.

Toute personne qui possède des autorisations de sécurité valides peut créer une URL pré-signée. Cependant, l'URL présignée doit être créée par une personne disposant des autorisations nécessaires pour effectuer l'opération sur laquelle l'URL présignée est basée. Pour plus d'informations, consultez Utilisateurs habilités à créer une URL présignée.

Utiliser les kits SDK AWS pour générer une URL présignée pour une opération d'objet S3 on Outposts

Java

SDK pour Java 2.x

Cet exemple montre comment générer une URL présignée que vous pouvez utiliser pour charger un objet dans un compartiment S3 on Outposts pour une durée limitée. Pour plus d'informations, consultez Utilisation d'URL présignée pour S3 on Outposts.

```
   public static void signBucket(S3Presigner presigner, String
outpostAccessPointArn, String keyName) {

       try {
           PutObjectRequest objectRequest = PutObjectRequest.builder()
                   .bucket(accessPointArn)
                   .key(keyName)
                   .contentType("text/plain")
```

```
                    .build();

            PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
                    .signatureDuration(Duration.ofMinutes(10))
                    .putObjectRequest(objectRequest)
                    .build();

            PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);


            String myURL = presignedRequest.url().toString();
            System.out.println("Presigned URL to upload a file to: " +myURL);
            System.out.println("Which HTTP method must be used when uploading a
file: " +
                    presignedRequest.httpRequest().method());

            // Upload content to the S3 on Outposts bucket by using this URL.
            URL url = presignedRequest.url();

            // Create the connection and use it to upload the new object by using
the presigned URL.
            HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
            connection.setDoOutput(true);
            connection.setRequestProperty("Content-Type","text/plain");
            connection.setRequestMethod("PUT");
            OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
            out.write("This text was uploaded as an object by using a presigned
URL.");
            out.close();

            connection.getResponseCode();
            System.out.println("HTTP response code is " +
connection.getResponseCode());

        } catch (S3Exception e) {
            e.getStackTrace();
        } catch (IOException e) {
            e.getStackTrace();
        }
```

```
    }
```

Python

SDK pour Python (Boto3)

Cet exemple montre comment générer une URL présignée qui peut exécuter une action S3 on Outposts pour une durée limitée. Pour plus d'informations, consultez Utilisation d'URL présignée pour S3 on Outposts. Pour effectuer une requête avec l'URL, utilisez le package `Requests`.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)


def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned S3 on Outposts URL that can be used to perform an
 action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds that the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
```

```
            "Couldn't get a presigned URL for client method '%s'.",
 client_method)
        raise
    return url


def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('-'*88)
    print("Welcome to the Amazon S3 on Outposts presigned URL demo.")
    print('-'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('accessPointArn', help="The name of the S3 on Outposts
 access point ARN.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in S3 on
 Outposts. For a "
                    "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()

    s3_client = boto3.client('s3')
    client_action = 'get_object' if args.action == 'get' else 'put_object'
    url = generate_presigned_url(
        s3_client, client_action, {'Bucket': args.accessPointArn, 'Key':
 args.key}, 1000)

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == 'get':
        response = requests.get(url)
    elif args.action == 'put':
        print("Putting data to the URL.")
        try:
            with open(args.key, 'r') as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
            print(f"Couldn't find {args.key}. For a PUT operation, the key must
 be the "
                  f"name of a file that exists on your computer.")
```

```
    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print('-'*88)


if __name__ == '__main__':
    usage_demo()
```

Amazon S3 sur les Outposts avec Amazon EMR local sur les Outposts

Amazon EMR est une plate-forme de cluster gérée qui simplifie l'exécution de structures de mégadonnées, telles que le traitement Apache Hadoop et Apache Spark AWS l'analyse de grandes quantités de données. En utilisant ces frameworks et les projets open source associés, vous pouvez traiter les données à des fins d'analyse et de charge de travail de business intelligence. Amazon EMR vous aide également à transformer et à déplacer de grandes quantités de données vers et depuis d'autres banques de données et bases de AWS données, et prend en charge Amazon S3 on Outposts. Pour plus d'informations sur Amazon EMR, consultez Amazon EMR on Outposts dans le guide de gestion Amazon EMR.

Pour Amazon S3 on Outposts, Amazon EMR a commencé à prendre en charge le connecteur Apache Hadoop S3A dans la version 7.0.0. Les versions antérieures d'Amazon EMR ne prennent pas en charge le S3 local sur Outposts, et le système de fichiers EMR (EMRFS) n'est pas pris en charge.

Applications prises en charge

Amazon EMR avec Amazon S3 on Outposts prend en charge les applications suivantes :

- Hadoop
- Spark
- Hue
- Hive
- Sqoop
- Pig
- Hudi

- Flink

Pour plus d'informations, consultez le Guide de version Amazon EMR.

Création et configuration d'un bucket Amazon S3 on Outposts

Amazon EMR utilise Amazon S3 on Outposts pour stocker les données d'entrée et de sortie. AWS SDK for Java Vos fichiers journaux Amazon EMR sont stockés dans un emplacement Amazon S3 régional que vous sélectionnez et ne sont pas stockés localement sur l'Outpost. Pour plus d'informations, consultez les journaux Amazon EMR dans le guide de gestion Amazon EMR.

Pour se conformer aux exigences d'Amazon S3 et du DNS, les compartiments S3 on Outposts sont soumis à des restrictions et limitations de dénomination. Pour plus d'informations, consultez Création d'un compartiment S3 on Outposts.

Avec Amazon EMR version 7.0.0 et versions ultérieures, vous pouvez utiliser Amazon EMR avec S3 sur Outposts et le système de fichiers S3A.

Prérequis

Autorisations S3 on Outposts — Lorsque vous créez votre profil d'instance Amazon EMR, votre rôle doit contenir l'espace de noms AWS Identity and Access Management (IAM) de S3 on Outposts. S3 on Outposts possède son propre espace de noms,. `s3-outposts*` Pour un exemple de politique utilisant cet espace de noms, consultezConfiguration d'IAM avec S3 on Outposts.

Connecteur S3A — Pour configurer votre cluster EMR afin d'accéder aux données d'un bucket Amazon S3 on Outposts, vous devez utiliser le connecteur S3A. Apache Hadoop Pour utiliser le connecteur, assurez-vous que tous vos URI S3 utilisent le `s3a` schéma. Si ce n'est pas le cas, vous pouvez configurer l'implémentation du système de fichiers que vous utilisez pour votre cluster EMR afin que vos URI S3 fonctionnent avec le connecteur S3A.

Pour configurer l'implémentation du système de fichiers afin qu'elle fonctionne avec le connecteur S3A, vous utilisez les propriétés de `fs.AbstractFileSystem.`*`file_scheme`*`.impl` configuration `fs.`*`file_scheme`*`.impl` et de votre cluster EMR, *`file_scheme`* qui correspondent au type d'URI S3 dont vous disposez. Pour utiliser l'exemple suivant, remplacez *`user input placeholders`* par vos propres informations. Par exemple, pour modifier l'implémentation du système de fichiers pour les URI S3 qui utilisent le `s3` schéma, spécifiez les propriétés de configuration de cluster suivantes :

```
[
  {
```

```
"Classification": "core-site",
    "Properties": {
    "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
    "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Pour utiliser S3A, définissez la propriété `fs.file_scheme.impl` de configuration sur`org.apache.hadoop.fs.s3a.S3AFileSystem`, puis définissez la `fs.AbstractFileSystem.file_scheme.impl` propriété sur. `org.apache.hadoop.fs.s3a.S3A`

Par exemple, si vous accédez au chemins`3a://bucket/...`, définissez la `fs.s3a.impl` propriété sur `org.apache.hadoop.fs.s3a.S3AFileSystem` et définissez la `fs.AbstractFileSystem.s3a.impl` propriété sur`org.apache.hadoop.fs.s3a.S3A`.

Commencer à utiliser Amazon EMR avec Amazon S3 sur Outposts

Les rubriques suivantes expliquent comment commencer à utiliser Amazon EMR avec Amazon S3 sur Outposts.

Rubriques

- Créer une stratégie d’autorisations
- Création et configuration de votre cluster
- Présentation des configurations
- Considérations

Créer une stratégie d’autorisations

Avant de créer un cluster EMR utilisant Amazon S3 on Outposts, vous devez créer une politique IAM à associer au profil d'instance Amazon EC2 du cluster. La politique doit disposer des autorisations nécessaires pour accéder au point d'accès Amazon Resource Name (ARN) S3 on Outposts. Pour plus d'informations sur la création de politiques IAM pour S3 on Outposts, consultez. Configuration d'IAM avec S3 on Outposts

L'exemple de politique suivant montre comment accorder les autorisations requises. Après avoir créé la politique, associez-la au rôle de profil d’instance que vous utilisez pour créer votre cluster EMR,

comme décrit dans la section the section called “Création et configuration de votre cluster”. Pour utiliser cet exemple, remplacez *user input placeholders* par vos propres informations.

```
{
"Version":"2012-10-17",
  "Statement": [
        {
  "Effect": "Allow",
            "Resource": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name,
            "Action": [
                "s3-outposts:*"
            ]
        }
    ]

 }
```

Création et configuration de votre cluster

Pour créer un cluster qui exécute Spark avec S3 sur Outposts, effectuez les étapes suivantes dans la console.

Pour créer un cluster qui s'exécute Spark avec S3 sur Outposts

1. Ouvrez la console Amazon EMR à l'adressehttps://console.aws.amazon.com/elasticmapreduce/.
2. Dans le panneau de navigation de gauche, choisissez Clusters.
3. Choisissez Créer un cluster.
4. Pour la version d'Amazon EMR, choisissez emr-7.0.0ou une version ultérieure.
5. Pour le pack d'applications, choisissez Spark interactive. Sélectionnez ensuite les autres applications prises en charge que vous souhaitez inclure dans votre cluster.
6. Pour activer Amazon S3 sur Outposts, entrez vos paramètres de configuration.

   Exemples de paramètres de configuration

   Pour utiliser les exemples de paramètres de configuration suivants, *user input placeholders* remplacez-les par vos propres informations.

   ```
   [
   ```

```
 {
   "Classification": "core-site",
   "Properties": {
     "fs.s3a.bucket.DOC-EXAMPLE-BUCKET.accesspoint.arn": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name"
     "fs.s3a.committer.name": "magic",
     "fs.s3a.select.enabled": "false"
    }
  },
  {
    "Classification": "hadoop-env",
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
          "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
          }
       }
     ],
     "Properties": {}
   },
   {
     "Classification": "spark-env",
     "Configurations": [
       {
         "Classification": "export",
         "Properties": {
           "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
         }
       }
      ],
      "Properties": {}
     },
     {
      "Classification": "spark-defaults",
      "Properties": {
        "spark.executorEnv.JAVA_HOME": "/usr/lib/jvm/java-11-amazon-
corretto.x86_64",
        "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
      }
     }
  ]
```

7. Dans la section Mise en réseau, choisissez un cloud privé virtuel (VPC) et un sous-réseau situés sur votre rack. AWS Outposts Pour plus d'informations sur Amazon EMR on Outposts, consultez la section Clusters EMR sur le manuel AWS Outposts Amazon EMR Management Guide.
8. Dans la section Profil d'instance EC2 pour Amazon EMR, choisissez le rôle IAM auquel est jointe la politique d'autorisation que vous avez créée précédemment.
9. Configurez les paramètres de cluster restants, puis choisissez Create cluster.

Présentation des configurations

Les tableaux suivants décrivent le S3A et les Spark configurations ainsi que les valeurs à spécifier pour leurs paramètres lorsque vous configurez un cluster qui utilise S3 sur des Outposts avec Amazon EMR.

Configurations du S3A

Paramètre	Valeur par défaut	Valeur requise pour S3 sur Outposts	Explication
`fs.s3a.aws.credentials.provider`	Si ce n'est pas spécifié, S3A recherchera S3 dans le compartiment Region avec le nom du compartiment Outposts.	L'ARN du point d'accès du bucket S3 on Outposts	Amazon S3 sur Outposts prend en charge les points d'accès Virtual Private Cloud (VPC) uniquement comme seul moyen d'accéder à vos compartiments Outposts.
`fs.s3a.committer.name`	`file`	`magic`	Magic Committer est le seul contributeur compatible pour S3 sur Outposts.
`fs.s3a.select.enabled`	`TRUE`	`FALSE`	S3 Select n'est pas pris en charge sur Outposts.

Paramètre	Valeur par défaut	Valeur requise pour S3 sur Outposts	Explication
`JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 sur Outposts sur S3A nécessite Java la version 11.

Configurations Spark

Paramètre	Valeur par défaut	Valeur requise pour S3 sur Outposts	Explication
`spark.sql.sources.fastS3PartitionDiscovery.enabled`	`TRUE`	`FALSE`	S3 sur Outposts ne prend pas en charge la partition rapide.
`spark.executorEnv.JAVA_HOME`	`/usr/lib/jvm/java-8`	`/usr/lib/jvm/java-11-amazon-corretto.x86_64`	S3 sur Outposts sur S3A nécessite la version 11 de Java.

Considérations

Tenez compte des points suivants lorsque vous intégrez Amazon EMR à S3 sur des compartiments Outposts :

- Amazon S3 on Outposts est compatible avec Amazon EMR version 7.0.0 et versions ultérieures.
- Le connecteur S3A est nécessaire pour utiliser S3 sur les Outposts avec Amazon EMR. Seul le S3A possède les fonctionnalités requises pour interagir avec S3 dans les compartiments Outposts. Pour obtenir des informations sur la configuration du connecteur S3A, reportez-vous à la section Conditions préalables.

- Amazon S3 on Outposts prend uniquement en charge le chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) avec Amazon EMR. Pour plus d'informations, consultez the section called "Chiffrement des données".
- Amazon S3 on Outposts ne prend pas en charge les écritures avec le S3A. FileOutputCommitter Les écritures effectuées avec le S3A FileOutputCommitter sur les compartiments S3 on Outposts entraînent l'erreur suivante InvalidStorageClass: La classe de stockage que vous avez spécifiée n'est pas valide.
- Amazon S3 on Outposts n'est pas compatible avec Amazon EMR Serverless ou Amazon EMR on EKS.
- Les journaux Amazon EMR sont stockés dans un emplacement Amazon S3 régional que vous sélectionnez, et ne sont pas stockés localement dans le compartiment S3 on Outposts.

Mise en cache des autorisations et des authentifications

S3 on Outposts met en cache de manière sécurisée les données d'authentification et d'autorisation localement sur les racks Outposts. Le cache supprime les allers-retours vers le parent Région AWS pour chaque demande. Cela élimine la variabilité introduite par les allers-retours du réseau. Avec le cache d'authentification et d'autorisation de S3 on Outposts, vous obtenez des latences constantes indépendantes de la latence de la connexion entre les Outposts et le. Région AWS

Lorsque vous faites une demande d'API S3 on Outposts, les données d'authentification et d'autorisation sont mises en cache de manière sécurisée. Les données mises en cache sont ensuite utilisées pour authentifier les demandes d'API d'objets S3 suivantes. S3 on Outposts met en cache les données d'authentification et d'autorisation uniquement lorsque la demande est signée à l'aide de Signature Version 4A (SigV4A). Le cache est stocké localement sur les Outposts au sein du service S3 on Outposts. Il est actualisé de manière asynchrone lorsque vous faites une demande d'API S3. Le cache est crypté et aucune clé cryptographique en texte brut n'est stockée sur Outposts.

Le cache est valide pendant 10 minutes maximum lorsque l'Outpost est connecté au Région AWS. Il est actualisé de manière asynchrone lorsque vous faites une demande d'API S3 on Outposts, afin de garantir que les dernières politiques sont utilisées. Si l'Outpost est déconnecté du Région AWS, le cache sera valide pendant 12 heures au maximum.

Configuration du cache d'autorisation et d'authentification

S3 on Outposts met automatiquement en cache les données d'authentification et d'autorisation pour les demandes signées avec l'algorithme Sigv4a. Pour plus d'informations, consultez la section

Signature des demandes d' AWS API dans le guide de AWS Identity and Access Management l'utilisateur. L'algorithme SigV4A est disponible dans les dernières versions des AWS SDK. Vous pouvez l'obtenir via une dépendance aux bibliothèques AWS Common Runtime (CRT).

Vous devez utiliser la dernière version du AWS SDK et installer la dernière version du CRT. Par exemple, vous pouvez exécuter `pip install awscrt` pour obtenir la dernière version du CRT avec Boto3.

S3 on Outposts ne met pas en cache les données d'authentification et d'autorisation pour les demandes signées avec l'algorithme SigV4.

Validation de la signature SigV4A

Vous pouvez l'utiliser AWS CloudTrail pour valider que les demandes ont été signées avec SIGv4a. Pour plus d'informations sur la configuration CloudTrail de S3 sur Outposts, consultez. Surveillance de S3 sur Outposts avec des journaux AWS CloudTrail

Après avoir configuré CloudTrail, vous pouvez vérifier comment une demande a été signée dans le `SignatureVersion` champ des CloudTrail journaux. Les demandes signées avec SIGv4a seront `SignatureVersion` définies sur. `AWS 4-ECDSA-P256-SHA256` Les demandes signées avec SigV4 seront `SignatureVersion` définies sur. `AWS 4-HMAC-SHA256`

Sécurité dans S3 on Outposts

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. En tant que client AWS, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entre AWS et vous. Le modèle de responsabilité partagée décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud – AWS est responsable de la protection de l'infrastructure qui exécute des Services AWS dans le AWS Cloud. AWS vous fournit également les services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des AWS programmes de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à Amazon S3 on Outposts, consultez Services AWS concernés par le programme de conformité .

- Sécurité dans le cloud – Votre responsabilité est fonction du Service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de S3 on Outposts. Les rubriques suivantes vous montrent comment configurer S3 on Outposts pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres Services AWS pour surveiller et sécuriser vos ressources S3 on Outposts.

Rubriques

Chiffrement des données dans S3 on Outposts

Par défaut, toutes les données stockées dans Amazon S3 on Outposts sont chiffrées à l'aide du chiffrement côté serveur avec les clés de chiffrement gérées Amazon S3 (SSE-S3). Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3).

Vous pouvez éventuellement utiliser le chiffrement côté serveur avec les clés de chiffrement fournies par le client (SSE-C). Pour utiliser SSE-C, spécifiez une clé de chiffrement dans le cadre de vos demandes d'API sur les objets. Un chiffrement côté serveur chiffre uniquement les données d'objet, pas les métadonnées d'objet. Pour plus d'informations, consultez Utilisation du chiffrement côté serveur avec les clés fournies par le client (SSE-C).

> **Note**
>
> S3 on Outposts ne prend pas en charge le chiffrement côté serveur avec les clés AWS Key Management Service (AWS KMS) (SSE-KMS).

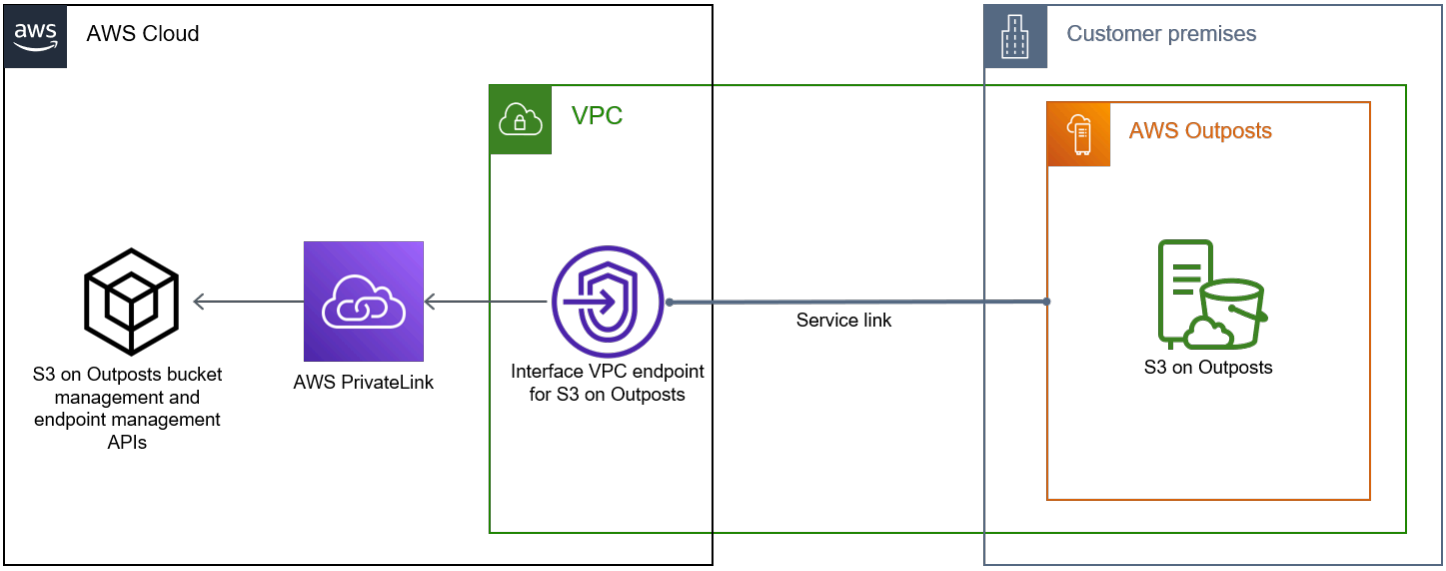
AWS PrivateLink pour S3 sur Outposts

S3 on Outposts prend en charge AWS PrivateLink, ce qui fournit un accès de gestion direct à votre stockage S3 on Outposts via un point de terminaison privé au sein de votre réseau privé virtuel. Cela vous permet de simplifier l'architecture de votre réseau interne et d'effectuer des opérations de gestion sur votre stockage d'objets Outposts en utilisant des adresses IP privées dans votre cloud privé virtuel (VPC). L'utilisation AWS PrivateLink élimine le besoin d'utiliser des adresses IP publiques ou des serveurs proxy.

Avec AWS PrivateLink Amazon S3 on Outposts, vous pouvez configurer des points de terminaison VPC d'interface dans votre cloud privé virtuel (VPC) pour accéder à vos API de gestion des compartiments et de gestion des points de terminaison S3 on Outposts. Les points de terminaison d'un VPC d'interface sont directement accessibles depuis des applications déployées dans votre VPC ou sur site par l'intermédiaire de votre réseau privé virtuel (VPN) ou d' AWS Direct Connect. Vous pouvez accéder aux API de gestion des compartiments et des terminaux via AWS PrivateLink. AWS PrivateLink ne prend pas en charge les opérations d'API de transfert de données, telles que GET, PUT et autres API similaires. Ces opérations sont déjà transférées en privé via la configuration du point de terminaison et du point d'accès S3 on Outposts. Pour plus d’informations, consultez Mise en réseau pour S3 on Outposts.

Les points de terminaison d'interface sont représentés par une ou plusieurs interfaces réseau Elastic (ENI) auxquelles des adresses IP privées sont attribuées à partir de sous-réseaux VPC. Les demandes envoyées à des points de terminaison d'interface pour S3 on Outposts sont automatiquement acheminées vers des API de gestion de compartiment et des points de terminaison S3 on Outposts sur le réseau AWS. Vous pouvez également accéder aux points de terminaison d'interface de votre VPC à partir d'applications sur site AWS Direct Connect via AWS Virtual Private Network ou ().AWS VPN Pour plus d'informations sur la façon de connecter votre VPC à votre réseau sur site, consultez le Guide de l'utilisateur AWS Direct Connect et le Guide de l'utilisateur AWS Site-to-Site VPN .

Les points de terminaison d'interface acheminent les demandes pour S3 sur le bucket Outposts et les API de gestion des points de terminaison via AWS le réseau et AWS PrivateLink via le réseau, comme illustré dans le schéma suivant.



Pour des informations générales sur les points de terminaison d'interface, consultez Points de terminaison de VPC d'interface (AWS PrivateLink) dans le Guide AWS PrivateLink .

Rubriques

- Restrictions et limitations
- Accès aux points de terminaison d'interface S3 on Outposts
- Mise à jour d'une configuration DNS sur site
- Création d'un point de terminaison de VPC pour S3 on Outposts
- Création de stratégies de compartiment et de stratégies de point de terminaison de VPC pour S3 on Outposts

Restrictions et limitations

Lorsque vous accédez à S3 on Outposts via le bucket et les API de gestion des terminaux, les limites des AWS PrivateLink VPC s'appliquent. Pour plus d'informations, consultez les sections Propriétés et limitations des points de terminaison d'interface et Quotas AWS PrivateLink du Guide AWS PrivateLink .

En outre, AWS PrivateLink ne prend pas en charge les éléments suivants :

- Points de terminaison FIPS (Federal Information Processing Standard)
- API de transfert de données S3 on Outposts, par exemple, GET, PUT et des opérations d'API d'objets similaires.

- DNS privé

Accès aux points de terminaison d'interface S3 on Outposts

Pour accéder au bucket S3 on Outposts et aux API de gestion des terminaux à l'aide des API de gestion des terminaux AWS PrivateLink, vous devez mettre à jour vos applications afin d'utiliser des noms DNS spécifiques aux points de terminaison. Lorsque vous créez un point de terminaison d'interface, deux types de S3 spécifiques au point de terminaison sont AWS PrivateLink générés sous les noms d'Outposts : régional et zonal.

- Noms DNS régionaux : incluez un ID de point de terminaison VPC unique, un identifiant de service, le Région AWS, et`vpce.amazonaws.com`, par exemple,. *`vpce-1a2b3c4d-5e6f`*`.s3-outposts.us-east-1.vpce.amazonaws.com`
- Noms DNS zonaux : incluez un ID de point de terminaison VPC unique, la zone de disponibilité, un identifiant de service, Région AWS le, `vpce.amazonaws.com` et, par exemple,. *`vpce-1a2b3c4d-5e6f`*`-us-east-1a.s3-outposts.us-east-1.vpce.amazonaws.com` Vous pouvez utiliser cette option si votre architecture isole les zones de disponibilité. Par exemple, vous pouvez utiliser des noms DNS de zone pour contenir les pannes ou réduire les coûts de transfert de données de région.

> **Important**
>
> Les points de terminaison de l'interface S3 on Outposts sont résolus depuis le domaine DNS public. S3 on Outposts ne prend pas en charge le DNS privé. Utilisez le paramètre `--endpoint-url` pour toutes les API de gestion des compartiments et des points de terminaison.

AWS CLI exemples

Utilisez les paramètres `--region` et `--endpoint-url` pour accéder aux API de gestion des compartiments et de gestion des points de terminaison via les points de terminaison d'interface S3 on Outposts.

Example : utiliser l'URL du point de terminaison pour répertorier les compartiments avec l'API de contrôle S3

Dans l'exemple suivant, remplacez la région *us-east-1*, l'URL de point de terminaison de VPC *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* et l'ID du compte *111122223333* par les informations appropriées.

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
 https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com --account-
id 111122223333
```

AWS Exemples de SDK

Mettez à jour vos kits SDK vers la dernière version et configurez vos clients pour qu'ils utilisent une URL de point de terminaison afin d'accéder à l'API de contrôle S3 pour les points de terminaison d'interface S3 on Outposts. Pour de plus amples informations, veuillez consulter Exemples de kit SDK AWS pour AWS PrivateLink.

SDK for Python (Boto3)

Example : utiliser une URL de point de terminaison pour accéder à l'API de contrôle S3

Dans l'exemple suivant, remplacez la région *us-east-1* et l'ID du point de terminaison de VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* par des informations appropriées.

```
control_client = session.client(
service_name='s3control',
region_name='us-east-1',
endpoint_url='https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com'
)
```

Pour plus d'informations, consultez AWS PrivateLink for Amazon S3 (pour Amazon S3) dans le Guide du développeur Boto3.

SDK for Java 2.x

Example : utiliser une URL de point de terminaison pour accéder à l'API de contrôle S3

Dans l'exemple suivant, remplacez l'URL de point de terminaison de VPC *vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com* et la région *Region.US_EAST_1* par des informations appropriées.

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://vpce-1a2b3c4d-5e6f.s3-outposts.us-
east-1.vpce.amazonaws.com"))
                                  .build()
```

Pour plus d'informations, consultez `S3ControlClient` dans la Référence d'API AWS SDK for Java .

Mise à jour d'une configuration DNS sur site

Lorsque vous utilisez des noms DNS spécifiques aux points de terminaison pour accéder aux points de terminaison d'interface pour les API de gestion de compartiments et de gestion de points de terminaison S3 on Outposts, vous n'avez pas besoin de mettre à jour votre résolveur DNS sur site. Vous pouvez résoudre le nom DNS spécifique au point de terminaison avec l'adresse IP privée du point de terminaison d'interface depuis le domaine DNS public S3 on Outposts.

Création d'un point de terminaison de VPC pour S3 on Outposts

Pour créer un point de terminaison d'interface de VPC pour S3 on Outposts, veuillez consulter Création d'un point de terminaison de VPC dans le Guide AWS PrivateLink .

Création de stratégies de compartiment et de stratégies de point de terminaison de VPC pour S3 on Outposts

Vous pouvez attacher une stratégie de point de terminaison à votre point de terminaison de VPC qui contrôle l'accès à S3 on Outposts. Vous pouvez également utiliser la condition `aws:sourceVpce` dans les stratégies de compartiment S3 on Outposts pour restreindre l'accès à des compartiments spécifiques depuis un point de terminaison de VPC spécifique. Avec les stratégies de point de terminaison de VPC, vous pouvez contrôler l'accès aux API de gestion des compartiments et aux API de gestion des points de terminaison S3 on Outposts. Avec les stratégies de compartiment, vous pouvez contrôler l'accès aux API de gestion des compartiments S3 on Outposts. Toutefois, vous ne pouvez pas gérer l'accès aux actions d'objet pour S3 on Outposts à l'aide de `aws:sourceVpce`.

Les stratégies d'accès pour S3 on Outposts spécifient les informations suivantes :

- Le principal AWS Identity and Access Management (IAM) pour lequel les actions sont autorisées ou refusées.
- Les actions de contrôle S3 qui sont autorisées ou refusées.
- Les ressources S3 on Outposts pour lesquelles des actions sont autorisées ou refusées.

Les exemples suivants montrent les stratégies qui restreignent l'accès à un compartiment ou à un point de terminaison. Pour plus d'informations sur la connectivité VPC, consultez la section Options de connectivité réseau à VPC dans le livre blanc Amazon AWS Virtual Private Cloud Connectivity Options.

> **Important**
>
> - Lors de l'application des exemples de stratégie pour les points de terminaison de VPC décrits dans cette section, vous pouvez bloquer involontairement votre accès au compartiment. Les autorisations attribuées à un compartiment qui restreignent l'accès aux connexions issues du point de terminaison de votre VPC peuvent bloquer toutes les connexions à ce compartiment. Pour des informations sur la correction de ce problème, veuillez consulter Ma politique de compartiment n'a pas le bon VPC ou ID de point de terminaison d'un VPC. Comment puis-je corriger la politique de façon à pouvoir accéder au compartiment ? que vous trouverez dans le AWS Support Centre de connaissances.
> - Avant d'utiliser l'exemple de stratégie de compartiment suivant, remplacez l'ID de point de terminaison de VPC par une valeur appropriée pour votre cas d'utilisation. Dans le cas contraire, vous ne parviendrez pas à accéder à votre compartiment.
> - Si votre stratégie n'autorise l'accès à un compartiment S3 on Outposts qu'à partir d'un point de terminaison de VPC spécifique, elle désactive l'accès à la console pour ce compartiment car les demandes de console ne proviennent pas du point de terminaison de VPC spécifié.

Rubriques

Exemple : restriction de l'accès à un compartiment spécifique depuis le point de terminaison d'un VPC

Vous pouvez créer une stratégie de point de terminaison qui restreint l'accès à des compartiments S3 on Outposts spécifiques uniquement. La politique suivante restreint l'accès à l' GetBucketPolicy action uniquement au*example-outpost-bucket*. Pour utiliser cette stratégie, remplacez les exemples de valeur par vos propres valeurs.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": {"AWS":"111122223333"},
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Allow",
      "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket"
    }
  ]
}
```

Exemple : refus d'accès depuis un point de terminaison de VPC spécifique dans une stratégie de compartiment S3 on Outposts

La politique de compartiment S3 on Outposts suivante refuse l'accès au *example-outpost-bucket* compartiment via GetBucketPolicy le point de terminaison VPC. *vpce-1a2b3c4d*

La condition `aws:sourceVpce` spécifie le point de terminaison et ne requiert pas d'Amazon Resource Name (ARN) pour la ressource de point de terminaison de VPC, mais uniquement l'ID du point de terminaison. Pour utiliser cette stratégie, remplacez les exemples de valeur par vos propres valeurs.

```
{
    "Version": "2012-10-17",
    "Id": "Policy1415115909152",
    "Statement": [
        {
            "Sid": "Deny-access-to-specific-VPCE",
            "Principal": {"AWS":"111122223333"},
            "Action": "s3-outposts:GetBucketPolicy",
```

```
            "Effect": "Deny",
            "Resource": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-
bucket",
            "Condition": {
                "StringEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}
            }
        }
    ]
}
```

Clés de stratégie spécifiques à l'authentification AWS Signature Version 4 (SigV4)

Le tableau suivant présente les clés de condition associées à l'authentification AWS Signature Version 4 (SigV4) que vous pouvez utiliser avec Amazon S3 on Outposts. Dans une stratégie de compartiment, vous pouvez ajouter ces conditions pour imposer un comportement spécifique lorsque les requêtes sont authentifiées en utilisant Signature Version 4. Pour obtenir des exemples de politiques, consultez Exemples de stratégies de compartiment qui utilisent des clés de condition associées à Signature Version 4. Pour de plus amples informations sur l'authentification des demandes en utilisant Signature Version 4, veuillez consulter la section Authentification des requêtes (AWS Signature Version 4) de la Référence des API Amazon Simple Storage Service

Clés applicables pour les actions **`s3-outposts:*`** ou l'une des actions S3 on Outposts

Clés applicables	Description
`s3-outposts:authType`	S3 on Outposts prend en charge différentes méthodes d'authentification. Pour limiter les requêtes entrantes afin d'utiliser d'une méthode d'authentification spécifique, vous pouvez utiliser cette clé de condition facultative. Par exemple, vous pouvez utiliser cette clé de condition pour autoriser uniquement l'en-tête HTTP`Authorization` pour l'authentification de la demande. Valeurs valides : `REST-HEADER` `REST-QUERY-STRING`

Clés applicables	Description
`s3-outposts:signatureAge`	La durée, en millisecondes, pendant laquelle une signature est valide dans une demande authentifiée. Cette condition ne fonctionne que pour les URL présignées. Dans Signature Version 4, la clé de signature est valide pendant sept jours au maximum. Par conséquent, les signatures ne restent valides que pendant sept jours. Pour de plus amples informations, veuillez consulter Introduction to signing requests (Introduction aux signatures des demandes) dans la Référence d'API Amazon Simple Storage Service. Vous pouvez utiliser cette condition pour limiter davantage la durée de la signature. Exemple de valeur : `600000`

Clés applicables	Description
`s3-outposts:x-amz-content-sha256`	Vous pouvez utiliser cette clé de condition pour interdire les contenus non signés dans votre compartiment. Lorsque vous utilisez Signature Version 4, pour les requêtes qui utilisent l'en-tête `Authorization` , vous ajoutez l'en-tête `x-amz-content-sha256` dans le calcul de signature, puis définissez sa valeur sur la charge utile du hachage. Vous pouvez utiliser cette clé de condition dans votre stratégie de compartiment pour refuser tous les chargements où les charges utiles ne sont pas signées. Par exemple : • Refuser les chargements qui utilisent l'en-tête `Authorization` pour authentifier les requêtes mais ne signent pas la charge utile. Pour de plus amples informations, veuillez consulter Transferring payload in a single chunk (Transfert de la charge utile en un seul fragment) dans la Référence d'API Amazon Simple Storage Service. • Refusez les chargements qui utilisent des URL présignées. Les URL présignées ont toujours une `UNSIGNED_PAYLOAD` . Pour de plus amples informations, veuillez consulter Authenticating requests (Requêtes d'authentification) et Authentication methods (Méthodes d'authentification dans la Référence d'API Amazon Simple Storage Service. Valeur valide : `UNSIGNED-PAYLOAD`

Exemples de stratégies de compartiment qui utilisent des clés de condition associées à Signature Version 4

Pour utiliser les exemples suivants, remplacez *`user input placeholders`* par vos propres informations.

Example : **s3-outposts:signatureAge**

La stratégie de compartiment suivante refuse toute demande d'URL présignée S3 on Outposts sur les objets dans `example-outpost-bucket` si la signature date de plus de 10 minutes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Deny a presigned URL request if the signature is more than 10
 minutes old",
            "Effect": "Deny",
            "Principal": {"AWS":"444455556666"},
            "Action": "s3-outposts:*",
            "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
            "Condition": {
                "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
                "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
            }
        }
    ]
}
```

Example : **s3-outposts:authType**

La stratégie de compartiment suivante autorise uniquement les requêtes qui utilisent l'en-tête `Authorization` pour l'authentification des demandes. Toute demande d'URL présignée sera refusée, car les URL présignées utilisent des paramètres de requête pour fournir des informations sur la requête et l'authentification. Pour de plus amples informations, veuillez consulter Authentication methods (Méthodes d'authentification dans la Référence d'API Amazon Simple Storage Service.

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
               "Sid": "Allow only requests that use the Authorization header for
 request authentication. Deny presigned URL requests.",
               "Effect": "Deny",
               "Principal": {"AWS":"111122223333"},
               "Action": "s3-outposts:*",
```

```
                "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
                "Condition": {
                      "StringNotEquals": {
                            "s3-outposts:authType": "REST-HEADER"
                      }
                }
          }
   ]
}
```

Example : **`s3-outposts:x-amz-content-sha256`**

La stratégie de compartiment suivante interdit les chargements avec des charges utiles non signées, tels que les chargements utilisant des URL présignées. Pour de plus amples informations, veuillez consulter Authenticating requests (Requêtes d'authentification) et Authentication methods (Méthodes d'authentification dans la Référence d'API Amazon Simple Storage Service.

```
{
   "Version": "2012-10-17",
   "Statement": [
         {
                "Sid": "Deny uploads with unsigned payloads.",
                "Effect": "Deny",
                "Principal": {"AWS":"111122223333"},
                "Action": "s3-outposts:*",
                "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/
*",
                "Condition": {
                      "StringEquals": {
                            "s3-outposts:x-amz-content-sha256": "UNSIGNED-PAYLOAD"
                      }
                }
          }
   ]
}
```

Politiques gérées AWS pour Amazon S3 sur Outposts

Une politique gérée par AWS est une politique autonome créée et administrée par AWS. Les politiques gérées par AWS sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

Gardez à l'esprit que les politiques gérées par AWS peuvent ne pas accorder les autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont disponibles pour tous les clients AWS. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les stratégies gérées par AWS. Si AWS met à jour les autorisations définies dans une politique gérée par AWS, la mise à jour affecte toutes les identités de principal (utilisateurs, groupes et rôles) auxquelles la politique est associée. AWS est plus susceptible de mettre à jour une politique gérée par AWS lorsqu'un nouveau Service AWS est lancé ou que de nouvelles opérations API deviennent accessibles pour les services existants.

Pour plus d'informations, consultez la rubrique Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

Politique gérée AWS : AWSS3OnOutpostsServiceRolePolicy

Gère les ressources réseau à votre place dans le cadre du rôle lié à un service `AWSServiceRoleForS3OnOutposts`.

Pour voir les autorisations que nécessite cette politique, consultez AWSS3OnOutpostsServiceRolePolicy.

Mises à jour de S3 sur Outposts dans les politiques gérées AWS

Obtenez des détails sur les mises à jour apportées aux politiques gérées AWS pour S3 sur Outposts depuis que ce service a commencé à suivre ces modifications.

Modification	Description	Date
Ajout de `AWSS3OnOutpostsServiceRolePolicy` dans S3 sur Outposts	S3 sur Outposts a ajouté `AWSS3OnOutpostsServiceRolePolicy` en tant	3 octobre 2023

Modification	Description	Date
	que partie intégrante du rôle lié à un service `AWSServiceRoleForS3OnOutposts` , qui gère les ressources réseau à votre place.	
Introduction du suivi des modifications dans S3 sur Outposts	S3 sur Outposts assure désormais le suivi des modifications pour ses politiques gérées AWS.	3 octobre 2023

Utilisation de rôles liés à un service pour Amazon S3 sur Outposts

Amazon S3 sur Outposts utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement à S3 sur Outposts. Les rôles liés à un service sont prédéfinis par S3 sur Outposts et englobent toutes les autorisations dont le service a besoin pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service simplifie la configuration de S3 sur Outposts, car vous n'avez pas besoin d'ajouter manuellement les autorisations nécessaires. S3 sur Outposts définit les autorisations de ses rôles liés à un service ; sauf définition contraire, seul S3 sur Outposts peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources S3 sur Outposts sont ainsi protégées, puisque vous ne pouvez pas supprimer accidentellement l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations de rôle lié à un service pour S3 sur Outposts

S3 sur Outposts utilise le rôle lié à un service nommé AWSServiceRoleForS3OnOutposts pour gérer les ressources réseau à votre place.

Le rôle lié à un service `AWSServiceRoleForS3OnOutposts` approuve les services suivants pour endosser le rôle :

- `s3-outposts.amazonaws.com`

La politique d'autorisations de rôle nommée `AWSS3OnOutpostsServiceRolePolicy` permet à S3 sur Outposts d'effectuer les actions suivantes sur les ressources spécifiées :

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeCoipPools",
                "ec2:GetCoipPoolUsage",
                "ec2:DescribeAddresses",
                "ec2:DescribeLocalGatewayRouteTableVpcAssociations"
            ],
            "Resource": "*",
            "Sid": "DescribeVpcResources"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ],
            "Sid": "CreateNetworkInterface"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface"
            ],
            "Resource": [
```

```
                "arn:aws:ec2:*:*:network-interface/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "CreateTagsForCreateNetworkInterface"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:ipv4pool-ec2/*"
            ],
            "Sid": "AllocateIpAddress"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AllocateAddress"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:elastic-ip/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "CreateTagsForAllocateIpAddress"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DisassociateAddress",
                "ec2:ReleaseAddress",
                "ec2:AssociateAddress"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/CreatedBy": "S3 On Outposts"
                }
            },
            "Sid": "ReleaseVpcResources"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "CreateNetworkInterface",
                        "AllocateAddress"
                    ],
                    "aws:RequestTag/CreatedBy": [
                        "S3 On Outposts"
                    ]
                }
            },
            "Sid": "CreateTags"
        }
    ]
}
```

Vous devez configurer des autorisations pour permettre à une entité IAM (comme un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour S3 sur Outposts

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un point de terminaison S3 sur Outposts dans la AWS Management Console, l'interface AWS CLI ou l'API AWS, S3 sur Outposts crée automatiquement le rôle lié à un service.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez un point de terminaison S3 sur Outposts, S3 sur Outposts recrée automatiquement le rôle lié à un service.

Vous pouvez également utiliser la console IAM pour créer un rôle lié à un service avec le cas d'utilisation S3 sur Outposts. Dans l'interface AWS CLI ou l'API AWS, créez un rôle lié à un service avec le nom de service `s3-outposts.amazonaws.com`. Pour de plus amples informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

Modification d'un rôle lié à un service pour S3 sur Outposts

S3 sur Outposts ne vous permet pas de modifier le rôle lié à un service `AWSServiceRoleForS30nOutposts`. Cela concerne également le nom du rôle, car diverses entités peuvent y faire référence. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour S3 sur Outposts

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

> **Note**
>
> Si le service S3 sur Outposts utilise le rôle pendant que vous tentez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources S3 sur Outposts utilisées par le rôle AWSServiceRoleForS3OnOutposts

1. Supprimez les points de terminaison S3 sur Outposts de votre Compte AWS dans toutes les Régions AWS.
2. Supprimez le rôle lié à un service à l'aide d'IAM.

Utilisez la console IAM, l'AWS CLI ou l'API AWS pour supprimer le rôle lié à un service `AWSServiceRoleForS3OnOutposts`. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service S3 sur Outposts

S3 sur Outposts prend en charge l'utilisation de rôles liés à un service dans toutes les Régions AWS où le service est disponible. Pour en savoir plus, consultez Régions et points de terminaison S3 sur Outposts.

Gestion de stockage S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour des applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour plus d'informations, consultez Qu'est-ce que Amazon S3 sur Outposts ?.

Pour plus d'informations sur la gestion et le partage de votre capacité de stockage Amazon S3 sur Outposts, consultez les rubriques suivantes.

Rubriques

- Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts
- Création et gestion d'une configuration de cycle de vie pour votre compartiment Amazon S3 on Outposts
- Répliquer des objets pour S3 sur Outposts
- Partage de S3 sur Outposts en utilisant AWS RAM
- Autre Services AWS utilisant S3 on Outposts

Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts

Une fois activé, la gestion des versions S3 enregistre plusieurs copies différentes d'un objet dans le même compartiment. Vous pouvez utiliser la gestion des versions S3 pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans votre compartiment Outposts. La gestion des versions S3 vous aide à récupérer en cas d'action involontaire d'un utilisateur et de défaillance applicative.

Les compartiments Amazon S3 on Outposts ont trois états de gestion des versions :

- Non versionné : si vous n'avez jamais activé ou suspendu la gestion des versions S3 sur votre compartiment, celle-ci est non versionnée et ne renvoie aucun statut de gestion des versions S3. Pour de plus amples informations sur la gestion des versions S3, veuillez consulter Utilisation de la gestion des versions dans les compartiments S3.
- Activé : active la gestion des versions S3 pour les objets du compartiment. Tous les objets ajoutés au compartiment reçoivent un ID de version unique. Les objets qui existaient déjà dans le compartiment au moment où vous activez la gestion des versions ont un ID de version égal à `null`. Si vous modifiez ces objets (ou tout autre) avec d'autres opérations, comme PutObject, les nouveaux objets reçoivent un ID de version unique.
- Suspendu : active la gestion des versions S3 pour les objets du compartiment. Tous les objets ajoutés au compartiment après que la gestion des versions ait été suspendue reçoivent l'ID de version `null`. Pour plus d'informations, consultez Ajout d'objets dans des compartiments désactivés pour la gestion des versions.

Lorsque vous activez la gestion des versions S3 pour un compartiment S3 on Outposts, il ne peut jamais revenir à un état non versionné. Toutefois, vous pouvez suspendre la gestion des versions. Pour de plus amples informations sur la gestion des versions S3, veuillez consulter Utilisation de la gestion des versions dans les compartiments S3.

Pour chaque objet de votre compartiment, vous avez une version actuelle et zéro, une ou plusieurs versions anciennes. Pour réduire les coûts de stockage, vous pouvez configurer les règles de cycle de vie de votre compartiment S3 de manière à ce que les versions anciennes expirent après une période spécifiée. Pour plus d'informations, consultez Création et gestion d'une configuration de cycle de vie pour votre compartiment Amazon S3 on Outposts.

Les exemples suivants vous montrent comment activer ou suspendre la gestion des versions pour un compartiment S3 on Outposts existant à l'aide de la AWS Management Console et de l'AWS Command Line Interface (AWS CLI). Pour créer un compartiment avec la gestion des versions S3 activée, consultez Création d'un compartiment S3 on Outposts.

> **Note**
>
> Le Compte AWS qui crée le compartiment en est le propriétaire et lui peut y valider des actions. Les compartiments possèdent des propriétés de configuration telles que Outpost, balise, chiffrement par défaut et paramètres de point d'accès. Les paramètres du point d'accès comprennent le cloud privé virtuel (VPC), la stratégie du point d'accès pour accéder aux objets du compartiment et d'autres métadonnées. Pour plus d'informations, consultez Spécifications de S3 on Outposts.

Utilisation de la console S3

Modifier les paramètres de la gestion des versions S3 pour votre compartiment

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts pour lequel vous voulez activer la gestion des versions S3.
4. Choisissez l'onglet Propriétés.
5. Sous Bucket Versioning (Gestion des versions de compartiment), choisissez Edit (Modifier).
6. Modifiez les paramètres de gestion des versions S3 pour le compartiment en choisissant l'une des options suivantes :
   - Pour suspendre la gestion des versions S3 et arrêter la création de nouvelles versions d'objets, choisissez Suspend (Suspendre).
   - Pour activer la gestion des versions S3 et enregistrer plusieurs copies distinctes de chaque objet, choisissez Enable (Activer).
7. Choisissez Enregistrer les modifications.

Utilisation de la AWS CLI

Pour activer ou suspendre la gestion des versions S3 pour votre compartiment à l'aide de l'interface AWS CLI, utilisez la commande `put-bucket-versioning`, comme indiqué dans les exemples suivants. Pour utiliser ces exemples, remplacez chaque *`user input placeholder`* par vos propres informations.

Pour plus d'informations, consultez put-bucket-versioning dans la Référence d'API AWS CLI.

Example : Activer la gestion des versions S3

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --versioning-configuration Status=Enabled
```

Example : Suspendre la gestion des versions S3

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --versioning-configuration Status=Suspended
```

Création et gestion d'une configuration de cycle de vie pour votre compartiment Amazon S3 on Outposts

Vous pouvez utiliser S3 Lifecycle pour optimiser la capacité de stockage d'Amazon S3 sur Outposts. Vous pouvez créer des règles de cycle de vie pour faire expirer les objets quand ils vieillissent ou sont remplacés par des versions plus récentes. Vous pouvez créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour plus d'informations sur le cycle de vie S3, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Le Compte AWS qui crée le compartiment en est le propriétaire et lui seul peut créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour créer et gérer la configuration du cycle de vie de votre compartiment S3 on Outposts, consultez les rubriques suivantes.

Rubriques

Création et gestion d'une règle de cycle de vie à l'aide de la AWS Management Console

Vous pouvez utiliser S3 Lifecycle pour optimiser la capacité de stockage d'Amazon S3 sur Outposts. Vous pouvez créer des règles de cycle de vie pour faire expirer les objets quand ils vieillissent ou sont remplacés par des versions plus récentes. Vous pouvez créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour plus d'informations sur le cycle de vie S3, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Le Compte AWS qui crée le compartiment en est le propriétaire et lui seul peut créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour créer et gérer une règle de cycle de vie pour un compartiment S3 sur Outposts à l'aide de la AWS Management Console, consultez les rubriques suivantes.

Rubriques

Création d'une stratégie de cycle de vie

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Sélectionnez le compartiment Outposts pour lequel vous voulez créer une règle de cycle de vie.

4. Choisissez l'onglet Management (Gestion), puis Create lifecycle rule (Créer une règle de cycle de vie).
5. Saisissez une valeur pour Lifecycle rule name (Nom de la règle de cycle de vie).
6. Sous Rule scope (Portée de la règle), choisissez l'une des options suivantes :

   - Pour limiter la portée à des filtres spécifiques, choisissez Limit the scope of this rule using one or more filters (Limiter la portée de cette règle en utilisant un ou plusieurs filtres). Ensuite, ajoutez un filtre de préfixe, des identifications ou une taille d'objet.
   - Pour appliquer la règle à tous les objets du compartiment, choisissez Apply to all objects in the bucket (Appliquer à tous les objets du compartiment).
7. Sous Lifecycle rule actions (Actions de règle de cycle de vie), choisissez l'une des options suivantes :

   - Expire current versions of objects (Faire expirer les versions actuelles des objets) : pour les compartiments compatibles avec la gestion des versions, S3 sur Outposts ajoute un marqueur de suppression et conserve les objets en tant que versions anciennes. Pour les compartiments qui n'utilisent pas la gestion des versions S3, S3 on Outposts supprime définitivement les objets.
   - Permanently delete noncurrent versions of objects (Supprimer définitivement les versions anciennes des objets) : S3 sur Outposts supprime définitivement les versions anciennes des objets.
   - Delete expired object delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression d'objet expirés ou les chargements partitionnés non terminés) : S3 sur Outposts supprime définitivement les marqueurs de suppression d'objet expirés ou les chargements partitionnés non terminés.

     Si vous limitez la portée de votre règle de cycle de vie en utilisant des balises d'objet, vous ne pouvez pas choisir Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés). Vous ne pouvez pas non plus choisir Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés) si vous choisissez Expire current object versions (Faire expirer les versions actuelles des objets).

> **Note**
>
> Les filtres basés sur la taille ne peuvent pas être utilisés avec des marqueurs de suppression et des chargements partitionnés non terminés.

8. Si vous avez choisi Expire current versions of objects (Faire expirer les versions actuelles des objets) ou Permanently delete noncurrent versions of objects (Supprimer définitivement les versions anciennes des objets), configurez le déclencheur de règles en fonction d'une date spécifique ou de l'âge de l'objet.
9. Si vous avez choisi Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés), pour confirmer que vous souhaitez supprimer les marqueurs de suppression d'objets expirés, sélectionnez Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés).
10. Sous Timeline Summary (Récapitulatif de la chronologie), vérifiez votre règle de cycle de vie et choisissez Create rule (Créer une règle).

Activer une stratégie de cycle de vie

Pour activer ou désactiver une règle de cycle de vie de compartiment

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts pour lequel vous voulez activer ou désactiver une règle de cycle de vie.
4. Choisissez l'onglet Management (Gestion), puis sous Lifecycle rule (Règle de cycle de vie), choisissez la règle que vous voulez activer ou désactiver.
5. Pour Action, choisissez Enable or disable rule (Activer ou désactiver la règle).

Modifier une stratégie de cycle de vie

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).

3. Choisissez le compartiment Outposts pour lequel vous voulez modifier une règle de cycle de vie.
4. Choisissez l'onglet Management (Gestion), puis la règle de cycle de vie que vous voulez modifier.
5. (Facultatif) Mettez à jour la valeur de Lifecycle rule name (Nom de la règle de cycle de vie).
6. Sous Rule scope (Portée de la règle), modifiez la portée selon vos besoins :

   - Pour limiter la portée à des filtres spécifiques, choisissez Limit the scope of this rule using one or more filters (Limiter la portée de cette règle en utilisant un ou plusieurs filtres). Ensuite, ajoutez un filtre de préfixe, des identifications ou une taille d'objet.
   - Pour appliquer la règle à tous les objets du compartiment, choisissez Apply to all objects in the bucket (Appliquer à tous les objets du compartiment).
7. Sous Lifecycle rule actions (Actions de règle de cycle de vie), choisissez l'une des options suivantes :

   - Expire current versions of objects (Faire expirer les versions actuelles des objets) : pour les compartiments compatibles avec la gestion des versions, S3 sur Outposts ajoute un marqueur de suppression et conserve les objets en tant que versions anciennes. Pour les compartiments qui n'utilisent pas la gestion des versions S3, S3 on Outposts supprime définitivement les objets.
   - Permanently delete noncurrent versions of objects (Supprimer définitivement les versions anciennes des objets) : S3 sur Outposts supprime définitivement les versions anciennes des objets.
   - Delete expired object delete markers or incomplete multipart uploads (Supprimer les marqueurs de suppression d'objet expirés ou les chargements partitionnés non terminés) : S3 sur Outposts supprime définitivement les marqueurs de suppression d'objet expirés ou les chargements partitionnés non terminés.

     Si vous limitez la portée de votre règle de cycle de vie en utilisant des balises d'objet, vous ne pouvez pas choisir Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés). Vous ne pouvez pas non plus choisir Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés) si vous choisissez Expire current object versions (Faire expirer les versions actuelles des objets).

> **Note**
>
> Les filtres basés sur la taille ne peuvent pas être utilisés avec des marqueurs de suppression et des chargements partitionnés non terminés.

8. Si vous avez choisi Expire current versions of objects (Faire expirer les versions actuelles des objets) ou Permanently delete noncurrent versions of objects (Supprimer définitivement les versions anciennes des objets), configurez le déclencheur de règles en fonction d'une date spécifique ou de l'âge de l'objet.
9. Si vous avez choisi Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés), pour confirmer que vous souhaitez supprimer les marqueurs de suppression d'objets expirés, sélectionnez Delete expired object delete markers (Supprimer les marqueurs de suppression d'objets expirés).
10. Choisissez Enregistrer.

Supprimer une stratégie de cycle de vie

1. Ouvrez la console Amazon S3 sur https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le compartiment Outposts pour lequel vous voulez supprimer une règle de cycle de vie.
4. Choisissez l'onglet Management (Gestion), puis sous Lifecycle rule (Règle de cycle de vie), choisissez la règle que vous voulez supprimer.
5. Sélectionnez Delete.

Création et gestion d'une configuration de cycle de vie à l'aide de l’AWS CLI et du SDK pour Java

Vous pouvez utiliser S3 Lifecycle pour optimiser la capacité de stockage d'Amazon S3 sur Outposts. Vous pouvez créer des règles de cycle de vie pour faire expirer les objets quand ils vieillissent ou sont remplacés par des versions plus récentes. Vous pouvez créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour plus d'informations sur le cycle de vie S3, consultez Gestion du cycle de vie de votre stockage.

> **Note**
>
> Le Compte AWS qui crée le compartiment en est le propriétaire et lui seul peut créer, activer, désactiver ou supprimer une règle de cycle de vie.

Pour créer et gérer une configuration de cycle de vie pour un compartiment S3 sur Outposts à l'aide de AWS Command Line Interface (AWS CLI) et AWS SDK for Java, consultez les exemples suivants.

Rubriques

- Exécution d'une commande de configuration PUT
- Exécution d'une commande de configuration de cycle de vie GET pour un compartiment S3 on Outposts

Exécution d'une commande de configuration PUT

AWS CLI

L'exemple d'utilisation de la AWS CLI suivant place une stratégie de configuration du cycle de vie sur un compartiment Outposts. Cette stratégie spécifie que tous les objets dont le préfixe est labelisé (*myprefix*), ainsi que les balises, expirent après 10 jours. Pour utiliser cet exemple, remplacez chaque *user input placeholder* par vos propres informations.

1. Enregistrez la stratégie de configuration du cycle de vie dans un fichier JSON. Dans cet exemple, le fichier est nommé `lifecycle1.json`.

```
{
    "Rules": [
        {
            "ID": "id-1",
            "Filter": {
                "And": {
                    "Prefix": "myprefix",
                    "Tags": [
                        {
                            "Value": "mytagvalue1",
                            "Key": "mytagkey1"
                        },
```

```
                        {
                            "Value": "mytagvalue2",
                            "Key": "mytagkey2"
                        }
                    ],
                    "ObjectSizeGreaterThan": 1000,
                    "ObjectSizeLessThan": 5000
                }
            },
            "Status": "Enabled",
            "Expiration": {
                "Days": 10
            }
        }
    ]
}
```

2. Envoyez le fichier JSON en tant que partie de la commande CLI `put-bucket-lifecycle-configuration`. Pour utiliser cette commande, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations sur cette commande, consultez put-bucket-lifecycle-configuration dans le document AWS CLI Reference.

```
aws s3control put-bucket-lifecycle-configuration --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket --lifecycle-configuration file://lifecycle1.json
```

SDK for Java

L'exemple d'utilisation de SDK pour Java suivant place une configuration de cycle de vie sur un compartiment Outposts. Cette configuration de cycle de vie spécifie que tous les objets dont le préfixe est labelisé (*`myprefix`*), ainsi que les balises, expirent après 10 jours. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations. Pour de plus amples informations, veuillez consulter PutBucketLifecycleConfiguration dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.services.s3control.model.*;

public void putBucketLifecycleConfiguration(String bucketArn) {

    S3Tag tag1 = new S3Tag().withKey("mytagkey1").withValue("mytagkey1");
```

```
    S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagkey2");

    LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
            .withAnd(new LifecycleRuleAndOperator()
                    .withPrefix("myprefix")
                    .withTags(tag1, tag2))
                    .withObjectSizeGreaterThan(1000)
                    .withObjectSizeLessThan(5000);

    LifecycleExpiration lifecycleExpiration = new LifecycleExpiration()
            .withExpiredObjectDeleteMarker(false)
            .withDays(10);

    LifecycleRule lifecycleRule = new LifecycleRule()
            .withStatus("Enabled")
            .withFilter(lifecycleRuleFilter)
            .withExpiration(lifecycleExpiration)
            .withID("id-1");

    LifecycleConfiguration lifecycleConfiguration = new LifecycleConfiguration()
            .withRules(lifecycleRule);

    PutBucketLifecycleConfigurationRequest reqPutBucketLifecycle = new
 PutBucketLifecycleConfigurationRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn)
            .withLifecycleConfiguration(lifecycleConfiguration);

    PutBucketLifecycleConfigurationResult respPutBucketLifecycle =
 s3ControlClient.putBucketLifecycleConfiguration(reqPutBucketLifecycle);
    System.out.printf("PutBucketLifecycleConfiguration Response: %s%n",
 respPutBucketLifecycle.toString());

}
```

Exécution d'une commande de configuration de cycle de vie GET pour un compartiment S3 on Outposts

AWS CLI

L'exemple d'utilisation de la AWS CLI suivant obtient une configuration de cycle de vie pour un compartiment Outposts. Pour utiliser cette commande, remplacez chaque *user input*

placeholder par vos propres informations. Pour de plus amples informations sur cette commande, consultez get-bucket-lifecycle-configuration dans le document AWS CLI Reference.

```
aws s3control get-bucket-lifecycle-configuration --account-id 123456789012 --bucket
 arn:aws:s3-outposts:<your-region>:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket
```

SDK for Java

L'exemple d'utilisation de SDK pour Java suivant obtient une configuration de cycle de vie pour un compartiment Outposts. Pour de plus amples informations, veuillez consulter GetBucketLifecycleConfiguration dans le document Amazon Simple Storage Service API Reference.

```
import com.amazonaws.services.s3control.model.*;

public void getBucketLifecycleConfiguration(String bucketArn) {

    GetBucketLifecycleConfigurationRequest reqGetBucketLifecycle = new
 GetBucketLifecycleConfigurationRequest()
            .withAccountId(AccountId)
            .withBucket(bucketArn);

    GetBucketLifecycleConfigurationResult respGetBucketLifecycle =
 s3ControlClient.getBucketLifecycleConfiguration(reqGetBucketLifecycle);
    System.out.printf("GetBucketLifecycleConfiguration Response: %s%n",
 respGetBucketLifecycle.toString());

}
```

Répliquer des objets pour S3 sur Outposts

Lorsque la réplication S3 est sur AWS Outposts, vous pouvez configurer Amazon S3 sur Outposts pour répliquer automatiquement les objets S3 entre différents Outposts ou entre des compartiments d'un même Outpost. Vous pouvez utiliser la réplication S3 sur Outposts pour gérer plusieurs réplicas de vos données dans le même Outpost ou dans des Outposts différents, ou sur différents comptes, afin de répondre aux besoins en matière de résidence des données. La réplication S3 sur Outposts permet de répondre à vos besoins de stockage conformes et de partager des données entre comptes. Si vous devez garantir que vos réplicas sont identiques aux données source, vous pouvez

utiliser la réplication S3 sur Outposts pour créer des réplicas de vos objets qui conservent toutes les métadonnées, telles que l'heure de création de l'objet d'origine, les balises et les ID de version.

La réplication S3 sur Outposts fournit également des métriques et des notifications détaillées pour surveiller le statut de la réplication des objets entre les compartiments. Vous pouvez utiliser Amazon CloudWatch pour surveiller la progression de la réplication en suivant les octets en attente de réplication, les opérations en attente de réplication et la latence de réplication entre vos compartiments source et de destination. Pour diagnostiquer et corriger rapidement les problèmes de configuration, vous pouvez également configurer Amazon EventBridge pour recevoir des notifications concernant les échecs des objets de réplication. Pour en savoir plus, veuillez consulter la section Gestion de votre réplication.

Rubriques

Configuration de réplication

S3 sur Outposts stocke une configuration de réplication au format XML. Dans le fichier XML de configuration de réplication, vous spécifiez un rôle AWS Identity and Access Management (IAM) et une ou plusieurs règles.

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

S3 sur Outposts ne peut pas répliquer d'objets sans votre autorisation. Vous accordez des autorisations S3 sur Outposts avec le rôle IAM que vous spécifiez dans la configuration de réplication. S3 sur Outposts endosse le rôle IAM pour répliquer des objets en votre nom. Vous devez accorder les autorisations requises au rôle IAM avant de commencer la réplication. Pour plus d'informations sur ces autorisations pour S3 sur Outposts, consultez Création d'un rôle IAM.

Vous ajoutez une règle dans la configuration de réplication pour les scénarios suivants :

- Vous souhaitez répliquer tous les objets.
- Vous souhaitez répliquer un sous-ensemble d'objets. Vous identifiez le sous-ensemble d'objets en ajoutant un filtre dans la règle. Dans le filtre, vous spécifiez un préfixe de clé d'objet et/ou des balises pour identifier le sous-ensemble d'objets auquel la règle s'applique.

Vous ajoutez plusieurs règles dans une configuration de réplication si vous souhaitez répliquer un sous-ensemble d'objets distinct. Dans chaque règle, vous spécifiez un filtre qui sélectionne un sous-ensemble d'objets différent. Par exemple, vous pouvez choisir de répliquer des objets qui possèdent les préfixes de clé `tax/` ou `document/`. Pour ce faire, vous ajoutez deux règles, l'une qui spécifie le filtre de préfixe de clé `tax/` et l'autre qui spécifie le préfixe de clé `document/`.

Pour plus d'informations sur la configuration et les règles de réplication de S3 sur Outposts, consultez ReplicationConfiguration (Configuration de la réplication) dans la Référence d'API Amazon Simple Storage Service.

Exigences pour la réplication S3 sur Outposts

La réplication exige de respecter les conditions suivantes :

- La plage d'adresses CIDR Outpost de destination doit être associée à votre table de sous-réseau Outpost source. Pour de plus amples informations, veuillez consulter Conditions préalables à la création de règles de réplication.
- La gestion des versions S3 doit être activée pour les compartiments source et de destination. Pour de plus amples informations sur la gestion des versions, veuillez consulter Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.
- Amazon S3 sur Outposts doit disposer des autorisations adéquates pour répliquer en votre nom les objets issus du compartiment source vers le compartiment de destination. Cela signifie que vous devez créer une fonction du service pour déléguer des autorisations `GET` et`PUT` à S3 sur Outposts.
  1. Avant de créer la fonction du service, vous devez disposer d'une autorisation `GET` sur le compartiment source et d'une autorisation `PUT` sur le compartiment de destination.

2. Pour créer la fonction du service permettant de déléguer des autorisations à S3 sur Outposts, vous devez d'abord configurer les autorisations afin de permettre à une entité IAM (utilisateur ou rôle) d'effectuer les actions `iam:CreateRole` et `iam:PassRole`. Enfin, vous autorisez l'entité IAM à créer une fonction du service. Afin de faire endosser à S3 sur Outposts la fonction du service en votre nom et déléguer des autorisations `GET` et `PUT` à S3 sur Outposts, vous devez attribuer les stratégies d'approbation et d'autorisation nécessaires au rôle. Pour plus d'informations sur ces autorisations pour S3 sur Outposts, consultez Création d'un rôle IAM. Pour plus d'informations sur la création d'une fonction du service, consultez Creating a service role (Création d'une fonction du service).

Ce qui est répliqué

Par défaut, S3 sur Outposts réplique les éléments suivants :

- Objets créés après l'ajout d'une configuration de réplication.
- Métadonnées d'objet des objets source vers les réplicas. Pour plus d'informations sur la réplication des métadonnées des réplicas vers les objets source, consultez Statut de la réplication si la synchronisation des modifications de réplica Amazon S3 sur Outposts est activée.
- Les balises d'objets, le cas échéant.

Impact des opérations de suppression sur la réplication

Si vous supprimez un objet du compartiment source, les actions suivantes se produisent par défaut :

- Si vous effectuez une demande `DELETE` sans spécifier d'ID de version d'objet, S3 sur Outposts ajoute un marqueur de suppression. S3 sur Outposts traite le marqueur de suppression comme suit :
  - S3 sur Outposts ne réplique pas le marqueur de suppression par défaut.
  - Toutefois, vous pouvez ajouter la réplication de marqueur de suppression aux règles non basées sur des balises. Pour plus d'informations sur la façon d'activer la réplication d'un marqueur de suppression dans votre configuration de réplication, consultez Utilisation de la console S3.
- Si vous spécifiez un ID de version d'objet à supprimer dans une demande `DELETE`, S3 sur Outposts supprime de façon permanente cette version de l'objet dans le compartiment source. Cependant, le service ne réplique pas la suppression dans les compartiments de destination. En d'autres termes, il ne supprime pas la même version de l'objet dans les compartiments de destination. Ce comportement protège les données des suppressions malveillantes.

Ce qui n'est pas répliqué

Par défaut, S3 sur Outposts ne réplique pas les éléments suivants :

- Les objets du compartiment source qui sont des réplicas ayant été créés par une autre règle de réplication. Supposons, par exemple, que vous configurez une réplication où le compartiment A est le compartiment source et le compartiment B celui de destination. Supposons ensuite que vous ajoutez une autre configuration de réplication où le compartiment B est le compartiment source et le compartiment C celui de destination. Dans ce cas, les objets du compartiment B qui sont les réplicas d'objets du compartiment A ne sont pas répliqués dans le compartiment C.
- Objets du compartiment source qui ont déjà été répliqués vers une autre destination. Par exemple, si vous changez le compartiment de destination dans une configuration de réplication existante, S3 sur Outposts ne procède pas à une nouvelle réplication des objets.
- Objets créés avec le chiffrement côté serveur à l'aide de clés de chiffrement fournies par le client (SSE-C).
- Les mises à jour des sous-ressources de niveau compartiment.

  Par exemple, si vous modifiez la configuration du cycle de vie ou ajoutez une configuration de notification à votre compartiment source, ces modifications ne sont pas appliquées au compartiment de destination. Cette fonction permet ainsi d'avoir des configurations différentes dans les compartiments source et de destination.
- Actions effectuées par la configuration du cycle de vie.

  Par exemple, si la configuration du cycle de vie est activée uniquement dans votre compartiment source et que vous configurez des actions d'expiration, S3 sur Outposts crée des marqueurs de suppression pour les objets expirés dans le compartiment source, mais ne réplique pas ces marqueurs dans les compartiments de destination. Si vous souhaitez appliquer la même configuration de cycle de vie aux compartiments source et de destination, activez la même configuration de cycle de vie sur les deux. Pour en savoir plus sur la configuration du cycle de vie, consultez Gestion du cycle de vie de votre stockage.

Qu'est-ce qui n'est pas pris en charge par la réplication S3 sur Outposts ?

Les fonctions de réplication S3 suivantes ne sont actuellement pas prises en charge par S3 sur Outposts :

- Contrôle du temps de réplication S3 (S3 RTC). S3 RTC n'est pas pris en charge car le trafic d'objets dans la réplication S3 sur Outposts passe par votre réseau sur site (la passerelle locale). Pour plus d'informations sur les passerelles locales, consultez Working with the local gateway (Utilisation des passerelles locales) dans le Guide de l'utilisateur AWS Outposts.
- Réplication S3 pour les opérations par lot.

Configuration de la réplication

> **Note**
>
> Les objets qui existaient dans votre compartiment avant la configuration de la règle de réplication ne sont pas répliqués automatiquement. En d'autres termes, Amazon S3 sur Outposts ne réplique pas les objets de manière rétroactive. Pour répliquer des objets créés avant la configuration de votre réplication, vous pouvez utiliser l'opération d'API `CopyObject` pour les copier dans le même compartiment. Une fois les objets copiés, ils apparaissent en tant que « nouveaux » objets dans le compartiment et la configuration de réplication s'appliquera à eux. Pour plus d'informations sur la copie d'un objet, consultez Copie d'un objet dans un compartiment Amazon S3 on Outposts à l'aide du kit AWS SDK for Java et CopyObject dans la Référence d'API Amazon Simple Storage Service.

Pour activer la réplication S3 sur Outposts, ajoutez une règle de réplication à votre compartiment Outposts source. La règle de réplication indique à S3 sur Outposts de répliquer les objets comme spécifié. Dans la règle de réplication, vous devez renseigner les éléments suivants :

- Point d'accès au compartiment Outposts source : Amazon Resource Name (ARN) du point d'accès ou alias du point d'accès du compartiment à partir duquel vous souhaitez que S3 sur Outposts réplique les objets S3. Pour plus d'informations sur l'utilisation des alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.
- Objets que vous voulez répliquer : vous pouvez répliquer l'ensemble ou un sous-ensemble des objets du compartiment Outposts source. Vous identifiez un sous-ensemble en fournissant un préfixe de nom de clé, une ou plusieurs balises d'objets, ou les deux dans la configuration.

  Par exemple, si vous configurez une règle de réplication pour ne répliquer que les objets dotés du préfixe de nom de clé `Tax/`, S3 sur Outposts réplique les objets avec des clés `Tax/doc1` et `Tax/doc2`. Mais le service ne réplique pas les objets dotés d'une clé `Legal/doc3`. Si vous spécifiez

un préfixe et une ou plusieurs balises, S3 sur Outposts réplique uniquement les objets dotés du préfixe de clé et des balises spécifiques.

- Compartiment Outposts de destination : ARN ou alias de point d'accès du compartiment vers lequel vous souhaitez que S3 sur Outposts réplique les objets.

Vous pouvez configurer la règle de réplication à l'aide de l'API REST, des kits SDK AWS, de l'AWS Command Line Interface (AWS CLI) ou de la console Amazon S3.

S3 sur Outposts fournit également des opérations d'API pour prendre en charge la configuration des règles de réplication. Pour de plus amples informations, consultez les rubriques suivantes dans la Référence d'API Amazon Simple Storage Service :

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Rubriques

- Conditions préalables à la création de règles de réplication
- Création de règles de réplication sur Outposts

Conditions préalables à la création de règles de réplication

Rubriques

- Connexion de vos sous-réseaux Outpost source et de destination
- Création d'un rôle IAM

Connexion de vos sous-réseaux Outpost source et de destination

Pour que votre trafic de réplication passe de votre Outpost source à votre Outpost de destination via votre passerelle locale, vous devez ajouter un nouvel acheminement pour configurer la mise en réseau. Vous devez connecter les plages de réseau de routage inter-domaines sans classe (CIDR) de vos points d'accès. Pour chaque paire de points d'accès, vous ne devez configurer cette connexion qu'une seule fois.

Certaines étapes de configuration de la connexion sont différentes, en fonction du type d'accès de vos points de terminaison Outposts associés à vos points d'accès. Le type d'accès pour les points de

terminaison est soit Privé (acheminement direct dans le cloud privé virtuel [VPC] pourAWS Outposts), soit Adresse IP détenue par le client (groupe d'adresses IP appartenant au client [groupe CoIP] au sein de votre réseau sur site).

Étape 1 : Trouver la plage CIDR de votre point de terminaison Outposts source

Pour trouver la plage CIDR de votre point de terminaison source associé à votre point d'accès source

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Dans la liste Compartiments Outposts, choisissez le compartiment source à partir duquel vous souhaitez effectuer la réplication.
4. Choisissez l'onglet Points d'accès Outposts, puis choisissez le point d'accès Outposts pour le compartiment source de votre règle de réplication.
5. Sélectionnez le point de terminaison Outposts.
6. Copiez l'ID de sous-réseau à utiliser à l'étape 5.
7. La méthode que vous utilisez pour trouver la plage d'adresses CIDR du point de terminaison Outposts source dépend du type d'accès de votre point de terminaison.

   Dans la section Présentation du point de terminaison Outposts, consultez Type d'accès.

   - Si le type d'accès est Privé, copiez la valeur CIDR (Classless Inter-Domain Routing) à utiliser à l'étape 6.
   - Si le type d'accès est Adresse IP détenue par le client, procédez comme suit :
     1. Copiez la valeur Groupe IPv4 appartenant au client pour l'utiliser ultérieurement comme ID du groupe d'adresses.
     2. Ouvrez la console AWS Outposts à l'adresse https://console.aws.amazon.com/outposts/.
     3. Dans le panneau de navigation, choisissez Tables de routage de passerelle locale.
     4. Choisissez la valeur ID de table de routage de passerelle locale de votre Outpost source.
     5. Dans le volet des détails, choisissez l'onglet Groupes CoIP. Collez la valeur de votre ID de groupe CoIP que vous avez copié précédemment dans le champ de recherche.
     6. Pour le groupe CoIP correspondant, copiez la valeur CIDR correspondante de votre point de terminaison Outposts source à l'utiliser à l'étape 6.

Étape 2 : Trouver l'ID de sous-réseau et la plage d'adresses CIDR de votre point de terminaison Outposts de destination

Pour trouver l'ID de sous-réseau et la plage d'adresses CIDR de votre point de terminaison de destination associé à votre point d'accès de destination, suivez les mêmes sous-étapes à l'étape 1 et remplacez votre point de terminaison Outposts source par votre point de terminaison Outposts de destination lorsque vous appliquez ces sous-étapes. Copiez la valeur de l'ID de sous-réseau de votre point de terminaison Outposts de destination pour l'utiliser à l'étape 6. Copiez la valeur CIDR de votre point de terminaison Outposts de destination pour l'utiliser à l'étape 5.

Étape 3 : Trouver l'ID de passerelle local de votre Outpost source

Pour trouver l'ID de passerelle local de votre Outpost source

1. Ouvrez la console AWS Outposts à l'adresse https://console.aws.amazon.com/outposts/.
2. Dans le volet de navigation de gauche, sélectionnez Passerelles locales.
3. Sur la page Passerelles locales, trouvez l'ID Outpost de votre Outpost source que vous souhaitez utiliser pour la réplication.
4. Copiez la valeur d'ID de passerelle locale de votre Outpost source pour l'utiliser à l'étape 5.

Pour plus d'informations sur les passerelles locales, consultez Local gateway (Passerelles locales) dans le Guide de l'utilisateur AWS Outposts.

Étape 4 : Trouver l'ID de passerelle local de votre Outpost de destination

Pour trouver l'ID de passerelle locale de votre Outpost de destination, suivez les mêmes sous-étapes qu'à l'étape 3, sauf que vous recherchez l'ID Outpost de votre Outpost de destination. Copiez la valeur d'ID de passerelle locale de votre Outpost de destination pour l'utiliser à l'étape 6.

Étape 5 : Configurer la connexion entre votre sous-réseau Outpost source et votre sous-réseau Outpost de destination

Pour connecter votre sous-réseau Outpost source et votre sous-réseau Outpost de destination

1. Connectez-vous à la AWS Management Console et ouvrez la console VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le panneau de navigation de gauche, choisissez Subnets (Sous-réseaux).

3. Dans la zone de recherche, entrez l'ID de sous-réseau de votre point de terminaison Outposts source que vous avez trouvé à l'étape 1. Choisissez le sous-réseau au sein de l'ID de sous-réseau correspondant.
4. Pour l'élément de sous-réseau correspondant, choisissez la valeur Table de routage de ce sous-réseau.
5. Sur la page contenant une table de routage sélectionnée, choisissez Actions, puis choisissez Modifier les routages.
6. Sur la page Modifier les routes, choisissez Ajouter une route.
7. Sous Destination, saisissez la plage d'adresses CIDR du point de terminaison Outposts de destination que vous avez trouvé à l'étape 2.
8. Sous Cible, choisissez Passerelle locale de l'Outpost et saisissez l'ID de passerelle locale de votre Outpost source que vous avez trouvé à l'étape 3.
9. Sélectionnez Save Changes (Enregistrer les modifications).
10. Assurez-vous que le Statut de la route est Actif.

Étape 6 : Configurer la connexion entre votre sous-réseau Outpost de destination et votre sous-réseau Outpost source

1. Connectez-vous à la AWS Management Console et ouvrez la console VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le panneau de navigation de gauche, choisissez Subnets (Sous-réseaux).
3. Dans la zone de recherche, saisissez l'ID de sous-réseau de votre point de terminaison Outposts de destination que vous avez trouvé à l'étape 2. Choisissez le sous-réseau au sein de l'ID de sous-réseau correspondant.
4. Pour l'élément de sous-réseau correspondant, choisissez la valeur Table de routage de ce sous-réseau.
5. Sur la page contenant une table de routage sélectionnée, choisissez Actions, puis choisissez Modifier les routages.
6. Sur la page Modifier les routes, choisissez Ajouter une route.
7. Sous Destination, saisissez la plage d'adresses CIDR du point de terminaison Outposts source que vous avez trouvé à l'étape 1.
8. Sous Cible, choisissez Passerelle locale de l'Outpost et saisissez l'ID de passerelle locale de votre Outpost de destination que vous avez trouvé à l'étape 4.

9. Sélectionnez Save Changes (Enregistrer les modifications).
10. Assurez-vous que le Statut de la route est Actif.

Après avoir connecté les plages de réseau CIDR de vos points d'accès source et de destination, vous devez créer un rôleAWS Identity and Access Management (IAM).

Création d'un rôle IAM

Par défaut, toutes les ressources S3 sur Outposts (compartiments, objets et sous-ressources liées) sont privées : seul le propriétaire des ressources peut y accéder. S3 sur Outposts a besoin d'autorisations pour lire et répliquer les objets du compartiment Outposts source. Vous accordez ces autorisations en créant une fonction du service IAM et en spécifiant ce rôle dans votre configuration de réplication.

Cette section décrit la stratégie d'approbation et la stratégie d'autorisation minimale requise. Les exemples de procédure fournissent des instructions étape par étape pour créer un rôle IAM. Pour de plus amples informations, veuillez consulter Création de règles de réplication sur Outposts. Pour plus d'informations sur les rôles IAM, consultez Rôles IAM dans le manuel IAM Guide de l'utilisateur.

- L'exemple suivant illustre une politique d'approbation selon laquelle vous identifiez S3 sur Outposts en tant que principal de service capable d'endosser le rôle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3-outposts.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- L'exemple suivant illustre une stratégie d'accès selon laquelle vous accordez au rôle les autorisations lui permettant d'effectuer les tâches de réplication en votre nom. Quand S3 sur Outposts endosse le rôle, il dispose des autorisations que vous avez spécifiées dans cette stratégie. Pour utiliser cette politique, remplacez *`user input placeholders`* par vos propres informations. Assurez-vous de les remplacer par les ID Outpost de vos Outposts source et de

destination, ainsi que par les noms des compartiments et des noms de points d'accès de vos compartiments Outposts source et de destination.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:GetObjectVersionForReplication",
            "s3-outposts:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:ReplicateObject",
            "s3-outposts:ReplicateDelete"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/
accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      }
   ]
}
```

La stratégie d'accès octroie les autorisations pour les actions suivantes :

- `s3-outposts:GetObjectVersionForReplication` : l'autorisation pour cette action est accordée sur tous les objets afin de permettre à S3 sur Outposts d'obtenir une version d'objet spécifique associée à chaque objet.

- `s3-outposts:GetObjectVersionTagging` : l'autorisation pour cette action sur des objets du compartiment *`SOURCE-OUTPOSTS-BUCKET`* (compartiment source) permet à S3 sur Outposts de lire les balises d'objets pour la réplication. Pour de plus amples informations, veuillez consulter Ajout de balises pour les compartiments Amazon S3 on Outposts. Si S3 sur Outposts ne dispose pas de ces autorisations, il réplique les objets mais pas leurs balises.
- `s3-outposts:ReplicateObject` et `s3-outposts:ReplicateDelete` : les autorisations pour ces actions sur tous les objets du compartiment *`DESTINATION-OUTPOSTS-BUCKET`* (compartiment cible) permettent à S3 sur Outposts de répliquer des objets ou des marqueurs de suppression dans le compartiment Outposts de destination. Pour de plus amples informations sur les marqueurs de suppression, consultez Impact des opérations de suppression sur la réplication.

> **Note**
>
> - L'autorisation pour l'action `s3-outposts:ReplicateObject` sur le compartiment *`DESTINATION-OUTPOSTS-BUCKET`* (compartiment de destination) permet également la réplication des balises d'objets. Il n'est donc pas nécessaire d'accorder une autorisation explicite pour l'action `s3-outposts:ReplicateTags`.
> - Pour la réplication intercompte, le propriétaire du compartiment Outposts de destination doit mettre à jour sa politique de compartiment afin d'autoriser l'action `s3-outposts:ReplicateObject` sur *`DESTINATION-OUTPOSTS-BUCKET`*. L'action `s3-outposts:ReplicateObject`permet à S3 sur Outposts de répliquer des objets et des balises d'objet vers le compartiment Outposts de destination.

Pour obtenir la liste des actions S3 sur Outposts, consultez Actions définies par Amazon S3 sur Outposts.

> **Important**
>
> Le Compte AWS qui possède le rôle IAM doit avoir des autorisations pour les actions qu'il octroie au rôle IAM.
> Supposons, par exemple, que le compartiment Outposts source contient des objets détenus par un autre Compte AWS. Le propriétaire des objets doit accorder de manière explicite au Compte AWS qui possède le rôle IAM les autorisations requises par l'intermédiaire de la politique de compartiment et de la politique de point d'accès. Dans le

cas contraire, S3 sur Outposts ne peut pas accéder aux objets et la réplication des objets échoue.
Les autorisations décrites dans la présente section sont liées à la configuration de réplication minimale. Si vous choisissez d'ajouter des configurations de réplication facultatives, vous devez accorder des autorisations supplémentaires à S3 sur Outposts.

Octroi d'autorisations lorsque les compartiments Outposts source et destination appartiennent à différents Comptes AWS

Lorsque les compartiments Outposts source et de destination n'appartiennent pas aux mêmes comptes, le propriétaire du compartiment Outposts de destination doit mettre à jour les politiques de compartiment et de point d'accès du compartiment de destination. Ces politiques doivent accorder au propriétaire du compartiment Outposts source et à la fonction du service IAM les autorisations nécessaires pour effectuer des actions de réplication, comme indiqué dans les exemples de politiques suivants, faute de quoi la réplication échouera. Dans ces exemples de politique, *DESTINATION-OUTPOSTS-BUCKET* est le compartiment de destination. Pour utiliser ces exemples de politique, remplacez *user input placeholders* par vos propres informations.

Si vous créez la fonction du service IAM manuellement, définissez le chemin du rôle sur `role/service-role/`, comme indiqué dans les exemples de politique suivants. Pour de plus amples informations, consultez ARN IAM dans le Guide de l'utilisateur IAM.

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
         ],
         "Resource":[
```

```
            "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*"
         ]
      }

   ]
}
```

```
{
"Version":"2012-10-17",
   "Id":"PolicyForDestinationAccessPoint",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::SourceBucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":[
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
         ],
         "Resource" :[
            "arn:aws:s3-outposts:region:DestinationBucket-account-
ID:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/
object/*"
         ]
      }
   ]
}
```

> **Note**
>
> Si des objets stockés dans le compartiment Outposts source sont balisés, notez les points suivants :
> Si le propriétaire du compartiment Outposts source octroie à S3 sur Outposts l'autorisation d'effectuer les actions `s3-outposts:GetObjectVersionTagging` et `s3-outposts:ReplicateTags` en vue de répliquer des balises d'objets (via le rôle IAM),

Amazon S3 réplique les balises en même temps que les objets. Pour obtenir des informations sur le rôle IAM, consultez Création d'un rôle IAM.

Création de règles de réplication sur Outposts

La réplication S3 sur Outposts est la copie automatique et asynchrone d'objets entre des compartiments dans des AWS Outposts différents ou identiques. Elle réplique les objets nouvellement créés et les mises à jour d'objets d'un compartiment Outposts source vers un ou plusieurs compartiments Outposts de destination. Pour de plus amples informations, veuillez consulter Répliquer des objets pour S3 sur Outposts.

> **Note**
>
> Les objets qui existaient dans votre compartiment Outposts source avant la configuration des règles de réplication ne sont pas répliqués. En d'autres termes, S3 sur Outposts ne réplique pas les objets de manière rétroactive. Pour répliquer des objets créés avant la configuration de votre réplication, vous pouvez utiliser l'opération d'API `CopyObject` pour les copier dans le même compartiment. Une fois les objets copiés, ils apparaissent en tant que « nouveaux » objets dans le compartiment et la configuration de réplication s'appliquera à eux. Pour plus d'informations sur la copie d'un objet, consultez Copie d'un objet dans un compartiment Amazon S3 on Outposts à l'aide du kit AWS SDK for Java et CopyObject dans la Référence d'API Amazon Simple Storage Service.

Lorsque vous configurez la réplication, vous ajoutez des règles de réplication au compartiment Outposts source. Les règles de réplication définissent les objets du compartiment Outposts source à répliquer, ainsi que le ou les compartiments Outposts de destination dans lesquels les objets répliqués seront stockés. Vous pouvez créer une règle pour répliquer tous les objets ou un sous-ensemble d'objets d'un compartiment à l'aide de préfixes de nom de clé ou d'autres balises d'objet, ou les deux. Un compartiment Outposts de destination peut se trouver dans le même Outpost que le compartiment Outposts source, mais il peut également se trouver dans un autre Outpost.

Pour les règles de réplication S3 sur Outposts, vous devez fournir à la fois l'Amazon Resource Name (ARN) du compartiment Outposts source et l'ARN du point d'accès du compartiment Outposts de destination au lieu des noms des compartiments Outposts source et de destination.

Si vous spécifiez un ID de version d'objet à supprimer, S3 sur Outposts supprime cette version de l'objet dans le compartiment Outposts source. Mais il ne réplique pas la suppression dans le

compartiment Outposts de destination. En d'autres termes, il ne supprime pas la même version de l'objet dans le compartiment Outposts de destination. Ce comportement protège les données des suppressions malveillantes.

Lorsque vous ajoutez une règle de réplication à un compartiment Outposts, celle-ci est activée par défaut et entre en fonctionnement dès que vous l'enregistrez.

Dans cet exemple, vous configurez la réplication pour les compartiments Outposts source et de destination qui sont sur différents Outposts et appartiennent au même Compte AWS. Des exemples sont fournis pour l'utilisation de la console Amazon S3, d'AWS Command Line Interface (AWS CLI), et des kits AWS SDK for Java et AWS SDK for .NET. Pour plus d'informations sur les autorisations de réplication S3 intercompte sur Outposts, consultez Octroi d'autorisations lorsque les compartiments Outposts source et destination appartiennent à différents Comptes AWS.

Pour connaître les conditions préalables aux règles de réplication S3 sur Outposts, consultez Conditions préalables à la création de règles de réplication.

Utilisation de la console S3

Suivez ces étapes pour configurer une règle de réplication quand le compartiment Amazon S3 sur Outposts de destination se trouve dans un Outpost différent de celui du compartiment Outposts source.

Si le compartiment Outposts de destination se trouve dans un compte différent du compartiment Outposts source, vous devez ajouter une stratégie de compartiment au compartiment Outposts de destination pour accorder au propriétaire du compte du compartiment Outposts source l'autorisation d'effectuer des réplications d'objets dans le compartiment Outposts de destination. Pour de plus amples informations, veuillez consulter Octroi d'autorisations lorsque les compartiments source et de destination appartiennent à des entités différentes Comptes AWS.

Pour créer une règle de réplication

1. Connectez-vous à la AWS Management Console et ouvrez la console Amazon S3 à l'adresse https://console.aws.amazon.com/s3/.
2. Dans la liste Compartiments Outposts, choisissez le nom du compartiment que vous voulez utiliser comme compartiment source.
3. Sélectionnez Gestion, faites défiler jusqu'à la section Règles de réplication, puis sélectionnez Créer une règle de réplication.

4. Sous Nom de la règle de réplication, saisissez un nom pour votre règle afin de l'identifier facilement plus tard. Ce nom est obligatoire et doit être unique dans le compartiment.
5. Sous Statut, Activé est sélectionné par défaut. Un règle activée entre en fonctionnement dès que l'avez enregistrée. Si vous souhaitez activer la règle ultérieurement, sélectionnez Désactivé.
6. Sous Priorité, la valeur de priorité de la règle détermine la règle à appliquer en cas de chevauchement de règles. Lorsque des objets sont inclus dans la portée de plusieurs règles de réplication, S3 sur Outposts utilise ces valeurs de priorité pour éviter les conflits. Par défaut, les nouvelles règles sont ajoutées à la configuration de la réplication avec la priorité la plus élevée. Plus le nombre est élevé, plus la priorité est haute.

   Pour modifier la priorité de la règle, après l'avoir enregistrée, choisissez le nom de la règle dans la liste des règles de réplication, choisissez Actions, puis Modifier la priorité.
7. Sous Compartiment source, vous disposez des options suivantes pour définir la source de réplication :

   - Pour répliquer l'ensemble du compartiment, choisissez Appliquer à tous les objets du compartiment.
   - Pour appliquer un filtrage par préfixe ou par balise à la source de réplication, choisissez Limiter la portée de cette règle en utilisant un ou plusieurs filtres. Vous pouvez combiner un préfixe et des balises.
     - Pour répliquer tous les objets ayant le même préfixe, sous Préfixe, entrez un préfixe dans la zone. Le filtre Préfixe permet de limiter la réplication à tous les objets dont le nom commence par la même chaîne (par exemple `pictures`).

       Si vous entrez un préfixe correspondant à un nom de dossier, vous devez insérer le caractère / (barre oblique) en tant que dernier caractère (par exemple, `pictures/`).
     - Pour répliquer tous les objets avec une ou plusieurs balises d'objet identiques, sélectionnez Ajouter une balise et saisissez la paire clé-valeur dans les zones. Pour ajouter une autre étiquette, répétez la procédure. Pour en savoir plus sur les balises d'objet, consultez Ajout de balises pour les compartiments Amazon S3 on Outposts.
8. Pour accéder à votre compartiment source S3 sur Outposts à des fins de réplication, sous Nom du point d'accès source, choisissez un point d'accès attaché au compartiment source.
9. Sous Destination, choisissez l'ARN du point d'accès du compartiment Outposts de destination dans lequel vous souhaitez que S3 sur Outposts réplique des objets. Le compartiment Outposts de destination peuvent se trouver dans différents Compte AWS ou dans les mêmes que le compartiment Outposts source.

Si le compartiment de destination se trouve dans un compte différent du compartiment Outposts source, vous devez ajouter une stratégie de compartiment au compartiment Outposts de destination pour accorder au propriétaire du compte du compartiment Outposts source l'autorisation d'effectuer des réplications d'objets dans le compartiment Outposts de destination. Pour de plus amples informations, veuillez consulter Octroi d'autorisations lorsque les compartiments Outposts source et destination appartiennent à différents Comptes AWS.

> **Note**
>
> Si la gestion des versions n'est pas activée sur le compartiment Outposts de destination, un message d'avertissement avec le bouton Activer la gestion des versions s'affiche. Cliquez sur ce bouton pour activer la gestion des versions sur le compartiment.

10. Configurez une fonction du service AWS Identity and Access Management (IAM) pouvant être endossé par S3 sur Outposts pour répliquer des objets en votre nom.

    Pour configurer un rôle IAM, sous Rôle IAM, effectuez l'une des opérations suivantes :

    - Pour que S3 sur Outposts crée un nouveau rôle IAM pour votre configuration de réplication, choisissez Choisir parmi les rôles IAM existants, puis Créer un nouveau rôle. Lorsque vous enregistrez la règle, une nouvelle stratégie est générée pour le rôle IAM correspondant aux compartiments Outposts source et cible que vous choisissez. Nous vous recommandons de choisir Créer un nouveau rôle.
    - Vous pouvez également choisir d'utiliser un rôle IAM existant. Dans ce cas, vous devez choisir un rôle qui octroie à S3 sur Outposts les autorisations nécessaires pour la réplication. La réplication échoue si ce rôle n'accorde pas à S3 sur Outposts des autorisations suffisantes pour suivre votre règle de réplication.

      Pour choisir un rôle existant, choisissez Choisir parmi les rôles IAM existants, puis choisissez le rôle dans le menu déroulant. Vous pouvez également choisir Saisir un ARN de rôle IAM, puis saisir l'Amazon Resource Name (ARN) du rôle.

    > **Important**
    >
    > Lorsque vous ajoutez une règle de réplication à un compartiment S3 sur Outposts, vous devez disposer des autorisations `iam:CreateRole` et `iam:PassRole` pour pouvoir créer et transmettre le rôle IAM qui accorde les autorisations de réplication S3 sur

Outposts. Pour plus d'informations, consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un Service AWS dans le Guide de l'utilisateur IAM.

11. Tous les objets contenus dans des compartiments Outposts sont chiffrés par défaut. Pour plus d'informations sur le chiffrement S3 sur Outposts, consultez la section Chiffrement des données dans S3 on Outposts. Seuls les objets chiffrés à l'aide du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3) peuvent être répliqués. La réplication d'objets chiffrés à l'aide du chiffrement côté serveur avec des clés AWS Key Management Service (AWS KMS) (SSE-KMS) ou du chiffrement côté serveur avec des clés fournies par le client (SSE-C) n'est pas prise en charge.
12. Au besoin, vous disposez des options supplémentaires suivantes lors de la définition de la configuration de la règle de réplication :

    - Si vous souhaitez activer les métriques de réplication S3 sur Outposts dans votre configuration de réplication, sélectionnez Métriques de réplication. Pour de plus amples informations, veuillez consulter Surveillance de la progression avec des métriques de réplication.
    - Si vous souhaitez activer la réplication de marqueurs de suppression dans votre configuration de réplication, sélectionnez Réplication des marqueurs de suppression. Pour de plus amples informations, veuillez consulter Impact des opérations de suppression sur la réplication.
    - Si vous souhaitez répliquer les modifications de métadonnées apportées aux réplicas vers les objets sources, sélectionnez Synchronisation des modifications de réplique. Pour de plus amples informations, veuillez consulter Statut de la réplication si la synchronisation des modifications de réplica Amazon S3 sur Outposts est activée.
13. Choisissez Créer une règle pour terminer.

Une fois que vous avez enregistré votre règle, vous pouvez la modifier, l'activer, la désactiver ou la supprimer. Pour ce faire, accédez à l'onglet Gestion du compartiment Outposts source, faites défiler la page jusqu'à la section Règles de réplication, choisissez votre règle, puis choisissez Modifier la règle.

Utilisation de AWS CLI

Pour utiliser AWS CLI afin de configurer une réplication quand les compartiments Outposts source et de destination appartiennent au même Compte AWS, effectuez les actions suivantes :

- Créez des compartiments Outposts source et de destination.
- Activez la gestion des versions sur les deux compartiments.

- Créez un rôle IAM qui octroie à S3 sur Outposts l'autorisation de répliquer des objets.
- Ajoutez la configuration de réplication au compartiment Outposts source.

Testez votre configuration pour la vérifier.

Pour configurer la réplication quand les compartiments Outposts source et de destination appartiennent au même Compte AWS

1. Définissez un profil d'informations d'identification pour l'AWS CLI. Dans cet exemple, nous utilisons le nom de profil `acctA`. Pour plus d'informations sur la définition des profils d'informations d'identification, consultez Named profiles (Profils nommés) dans le Guide de l'utilisateur AWS Command Line Interface.

   > **Important**
   >
   > Le profil que vous utilisez dans cet exercice doit disposer des autorisations nécessaires. Par exemple, dans la configuration de la réplication, vous spécifiez la fonction du service IAM que S3 sur Outposts peut endosser. Vous ne pouvez effectuer cette tâche que si le profil que vous utilisez dispose des autorisations `iam:CreateRole` et `iam:PassRole`. Pour plus d'informations, consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un Service AWS dans le Guide de l'utilisateur IAM. Si vous utilisez les informations d'identification d'un administrateur pour créer un profil nommé, le profil nommé disposera des autorisations adéquates pour exécuter toutes les tâches.

2. Créez un compartiment *`source`* et activez le contrôle de version sur ce dernier. La commande `create-bucket` suivante crée un compartiment *`SOURCE-OUTPOSTS-BUCKET`* dans la région USA Est (Virginie du Nord) (`us-east-1`). Pour utiliser cette commande, remplacez *`user input placeholders`* par vos propres informations.

   ```
   aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
   ```

   La commande `put-bucket-versioning` suivante active la gestion des versions sur le compartiment *`SOURCE-OUTPOSTS-BUCKET`*. Pour utiliser cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

3. Créez un compartiment de *destination* et activez le contrôle de version sur ce dernier. La commande `create-bucket` suivante crée un compartiment *DESTINATION-OUTPOSTS-BUCKET* dans la région USA Ouest (Oregon) (`us-west-2`). Pour utiliser cette commande, remplacez *user input placeholders* par vos propres informations.

   > **Note**
   >
   > Pour procéder à la configuration de la réplication lorsque les compartiments Outposts source et de destination appartiennent au même Compte AWS, vous utilisez le même profil nommé. Cet exemple utilise `acctA`. Pour tester la configuration de la réplication lorsque les compartiments sont détenus par des Comptes AWS différents, vous spécifiez des profils différents pour chaque compartiment.

   ```
   aws s3control create-bucket --bucket DESTINATION-OUTPOSTS-BUCKET --create-bucket-configuration LocationConstraint=us-west-2 --outpost-id DESTINATION-OUTPOST-ID --profile acctA --region us-west-2
   ```

   La commande `put-bucket-versioning` suivante active la gestion des versions sur le compartiment *DESTINATION-OUTPOSTS-BUCKET*. Pour utiliser cette commande, remplacez *user input placeholders* par vos propres informations.

   ```
   aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
   ```

4. Créez une fonction du service IAM. Vous ajouterez cette fonction du service au compartiment *SOURCE-OUTPOSTS-BUCKET* plus tard dans la configuration de la réplication. S3 sur Outposts endosse ce rôle pour répliquer des objets en votre nom. Vous créez un rôle IAM en deux étapes.

   a. Créez un rôle IAM.

      i. Copiez la stratégie d'approbation suivante et enregistrez-la dans un fichier nommé `s3-on-outposts-role-trust-policy.json` dans le répertoire actif sur votre

ordinateur local. Cette stratégie accorde au principal de service S3 sur Outposts les autorisations pour endosser cette fonction du service.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3-outposts.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

ii. Exécutez la commande suivante pour créer le rôle. Remplacez *user input placeholders* par vos propres informations.

```
aws iam create-role --role-name replicationRole --assume-role-policy-
document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

b. Attachez une stratégie d'autorisation à la fonction du service.

i. Copiez la politique d'autorisations suivante et enregistrez-la dans un fichier nommé `s3-on-outposts-role-permissions-policy.json` dans le répertoire actuel de votre ordinateur local. Cette stratégie accorde des autorisations pour diverses actions sur les compartiments et les objets S3 sur Outposts. Pour utiliser cette politique, remplacez *user input placeholders* par vos propres informations.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:GetObjectVersionForReplication",
            "s3-outposts:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
```

```
            "arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-
OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3-outposts:ReplicateObject",
            "s3-outposts:ReplicateDelete"
         ],
         "Resource":[
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
            "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-
OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
         ]
      }
   ]
}
```

ii. Exécutez la commande suivante pour créer une stratégie et l'attacher au rôle. Remplacez *user input placeholders* par vos propres informations.

```
aws iam put-role-policy --role-name replicationRole --policy-
document file://s3-on-outposts-role-permissions-policy.json --policy-
name replicationRolePolicy --profile acctA
```

5. Ajoutez une configuration de réplication au compartiment *SOURCE-OUTPOSTS-BUCKET*.

   a. Même si l'API S3 sur Outposts nécessite une configuration de la réplication au format XML, l'AWS CLI requiert que vous spécifiiez la configuration de réplication au format JSON. Enregistrez la configuration JSON dans un fichier (`replication.json`) dans le répertoire local de votre ordinateur. Pour utiliser cette configuration, remplacez *user input placeholders* par vos propres informations.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
```

```
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket":
        "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-
ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT"
      }
    }
  ]
}
```

b. Pour ajouter la configuration de réplication à votre compartiment Outposts source, exécutez la commande `put-bucket-replication` suivante. Pour utiliser cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control put-bucket-replication --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-
ID/bucket/SOURCE-OUTPOSTS-BUCKET --replication-configuration file://
replication.json --profile acctA
```

c. Pour récupérer la configuration de réplication, utilisez la commande `get-bucket-replication`. Pour utiliser cette commande, remplacez *`user input placeholders`* par vos propres informations.

```
aws s3control get-bucket-replication --account-id 123456789012 --bucket
 arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET --profile acctA
```

6. Testez la configuration dans la console Amazon S3 :

   a. Connectez-vous à la AWS Management Console et ouvrez la console Simple Storage Service (Amazon S3) à la page https://console.aws.amazon.com/s3/.

   b. Dans le compartiment *`SOURCE-OUTPOSTS-BUCKET`*, créez un dossier nommé `Tax`.

   c. Ajoutez les exemples d'objets au dossier `Tax` du compartiment *`SOURCE-OUTPOSTS-BUCKET`*.

   d. Dans le compartiment *`DESTINATION-OUTPOSTS-BUCKET`*, vérifiez les éléments suivants :

      - S3 sur Outposts a répliqué les objets.

> **Note**
>
> Le temps nécessaire à S3 sur Outposts pour répliquer un objet dépend de la taille de ce dernier. Pour obtenir des informations sur la consultation du statut de la réplication, consultez Obtention d'informations sur le statut de la réplication.

- Dans l'onglet Propriétés, le Statut de réplication est défini sur Réplica (afin d'indiquer qu'il s'agit d'un objet réplica).

Gestion de votre réplication

Cette section décrit des options de configuration de réplication supplémentaires disponibles dans S3 sur Outposts, comment déterminer le statut de la réplication et comment résoudre des problèmes de réplication. Pour obtenir des informations sur la configuration de réplication de base, veuillez consulter Configuration de la réplication.

Rubriques

- Surveillance de la progression avec des métriques de réplication
- Obtention d'informations sur le statut de la réplication
- Résolution des problèmes de réplication
- Utilisation d'EventBridge pour la réplication S3 sur Outposts

Surveillance de la progression avec des métriques de réplication

La réplication S3 sur Outposts fournit des métriques détaillées pour les règles de réplication dans votre configuration de réplication. Avec les métriques de réplication, vous pouvez surveiller la progression de la réplication par intervalles de 5 minutes en suivant les octets en attente, la latence de réplication et les opérations en attente. Pour vous aider à résoudre les problèmes de configuration, vous pouvez également configurer Amazon EventBridge afin qu'il reçoive des notifications sur les échecs de réplication.

Lorsque les métriques de réplication sont activées, la réplication S3 sur Outposts publie les métriques suivantes sur Amazon CloudWatch :

- Octets en attente de réplication : nombre total d'octets d'objets en attente de réplication pour une règle de réplication donnée.

- Latence de réplication : nombre maximal de secondes pendant lesquelles le compartiment de destination de réplication se trouve derrière le compartiment source pour une règle de réplication donnée.
- Opérations en attente de réplication : nombre d'opérations en attente de réplication pour une règle de réplication donnée. Les opérations incluent des objets, des marqueurs de suppression et des balises.

> **Note**
>
> Les métriques de réplication S3 sur Outposts sont facturées au même tarif que les métriques CloudWatch personnalisées. Pour en savoir plus , consultez Tarification CloudWatch.

Obtention d'informations sur le statut de la réplication

Le statut de réplication peut vous aider à déterminer l'état actuel d'un objet répliqué par Amazon S3 sur Outposts. Le statut de réplication d'un objet source renvoie soit `PENDING`, `COMPLETED` ou `FAILED`. Le statut de réplication d'un réplica renvoie `REPLICA`.

Vue d'ensemble des statuts de réplication

Dans un scénario de réplication, il existe un compartiment source dans lequel vous configurez la réplication et un compartiment de destination dans lequel S3 sur Outposts réplique les objets. Lorsque vous demandez un objet (avec `GetObject`) ou des métadonnées d'objet (avec `HeadObject`) à partir de ces compartiments, S3 sur Outposts renvoie l'en-tête `x-amz-replication-status` dans la réponse, comme suit :

- Lorsque vous demandez un objet depuis le compartiment source, S3 sur Outposts renvoie l'en-tête `x-amz-replication-status` si l'objet demandé peut être répliqué.

  Par exemple, imaginons que vous spécifiez le préfixe d'objet `TaxDocs` dans votre configuration de réplication pour indiquer à S3 sur Outposts de ne répliquer que les objets dotés du préfixe de nom de clé `TaxDocs`. Tous les objets que vous chargez ayant ce préfixe de nom de clé (par exemple, `TaxDocs/document1.pdf`) seront répliqués. Pour les demandes d'objet avec ce préfixe de nom de clé, S3 sur Outposts renvoie l'en-tête `x-amz-replication-status` avec l'une des valeurs suivantes pour le statut de réplication de l'objet : `PENDING`, `COMPLETED` ou `FAILED`.

> **Note**
>
> Si la réplication d'objet échoue après avoir chargé un objet, vous ne pouvez pas relancer la réplication. Vous devez recharger l'objet. Les objets passent à l'état `FAILED` en cas de problèmes, par exemple si les autorisations de rôle de réplication ou les autorisations de compartiment sont manquantes. Pour les échecs temporaires, par exemple si un compartiment ou votre Outpost n'est pas disponible, le statut de réplication ne passera pas à `FAILED`, mais restera `PENDING`. Une fois la ressource remise en ligne, S3 sur Outposts reprendra la réplication de ces objets.

- Lorsque vous demandez un objet à partir du compartiment de destination, si l'objet demandé est un réplica créé par S3 sur Outposts, S3 sur Outposts renvoie l'en-tête `x-amz-replication-status` avec la valeur `REPLICA`.

> **Note**
>
> Avant de supprimer un objet d'un compartiment source pour lequel la réplication a été activée, contrôlez le statut de réplication de l'objet pour vérifier qu'il a été répliqué.

Statut de la réplication si la synchronisation des modifications de réplica Amazon S3 sur Outposts est activée

Lorsque vos règles de réplication activent la synchronisation des modifications de réplica S3 sur Outposts, les statuts des réplicas peuvent être différents de `REPLICA`. Si des modifications de métadonnées sont en cours de réplication, l'en-tête `x-amz-replication-status` du réplica renvoie `PENDING`. Si la synchronisation des modifications du réplica ne parvient pas à répliquer les métadonnées, l'en-tête du réplica renvoie `FAILED`. Si les métadonnées sont répliquées correctement, l'en-tête du réplica renvoie la valeur `REPLICA`.

Résolution des problèmes de réplication

Si les réplicas d'objets ne figurent pas dans le compartiment Amazon S3 sur Outposts de destination après la configuration de la réplication, utilisez les conseils de dépannage suivants pour identifier les problèmes et les résoudre.

- Le temps nécessaire à S3 sur Outposts pour répliquer un objet dépend de plusieurs facteurs, y compris de la distance entre la paire de régions source et de destination et de la taille de l'objet.

  Vous pouvez vérifier le statut de réplication de l'objet source. Si le statut de réplication de l'objet est `PENDING`, cela signifie que S3 sur Outposts n'a pas terminé la réplication. Si le statut de réplication de l'objet est `FAILED`, vérifiez la configuration de réplication définie sur le compartiment source.
- Dans la configuration de réplication du compartiment source, procédez aux vérifications suivantes :
  - L'Amazon Resource Name (ARN) du point d'accès du compartiment de destination est correct.
  - Le préfixe de nom de clé est correct. A titre d'exemple, si vous définissez la configuration pour répliquer des objets avec le préfixe `Tax`, seuls les objets dotés de noms de clés `Tax/document1` ou `Tax/document2` seront répliqués. Tout objet avec le nom de clé `document3` n'est pas répliqué.
  - Le statut est `Enabled`.
- Vérifiez que la gestion des versions n'a pas été suspendue sur aucun compartiment. La gestion des versions doit être activée pour les compartiments source et de destination.
- Si le compartiment de destination appartient à un autre Compte AWS, vérifiez si le propriétaire du compartiment dispose d'une stratégie de compartiment sur le compartiment de destination, qui permet au propriétaire du compartiment source de répliquer des objets. Pour voir un exemple, consultez Octroi d'autorisations lorsque les compartiments Outposts source et destination appartiennent à différents Comptes AWS.
- Si un réplica d'objet ne figure pas dans le compartiment de destination, les problèmes suivants ont pu empêcher la réplication :
  - S3 sur Outposts ne réplique pas un objet figurant dans un compartiment source qui est lui-même un réplica créé par une autre configuration de réplication. Par exemple, si vous définissez une configuration de réplication à partir du compartiment A vers le compartiment B vers le compartiment C, S3 sur Outposts ne réplique pas les réplicas d'objets dans le compartiment B vers le compartiment C.

    Si vous souhaitez répliquer des objets du compartiment A vers le compartiment B et le compartiment C, définissez plusieurs destinations de compartiment selon différentes règles de réplication pour la configuration de la réplication de votre compartiment source. Par exemple, créez deux règles de réplication sur le compartiment source A, l'une pour la réplication vers le compartiment de destination B et l'autre pour la réplication vers le compartiment de destination C.

  - Un propriétaire de compartiment source peut accorder à d'autres Comptes AWS l'autorisation de charger des objets. Par défaut, le propriétaire du compartiment source ne possède aucune autorisation pour les objets créés par d'autres comptes. La configuration de réplication réplique uniquement les objets pour lesquels le propriétaire du compartiment source dispose des autorisations d'accès. Pour éviter les problèmes de réplication, le propriétaire du compartiment source peut accorder d'autres autorisations Comptes AWS permettant de créer des objets de manière conditionnelle, en exigeant des autorisations d'accès explicites sur ces objets. Pour un exemple de politique, consultez Octroi d'autorisations intercomptes pour charger des objets tout en garantissant que le propriétaire du compartiment dispose d'un contrôle total.
- Supposons que vous ajoutiez une règle dans la configuration de réplication pour répliquer un sous-ensemble d'objets dotés d'une balise spécifique. Dans ce cas, vous devez attribuer une clé et une valeur de balise spécifiques au moment de la création de l'objet pour que S3 sur Outposts puisse répliquer l'objet. Si vous commencez par créer un objet, puis ajoutez la balise à l'objet existant, S3 sur Outposts ne réplique pas l'objet.
- La réplication échoue si la stratégie de compartiment refuse l'accès au rôle de réplication pour l'une des actions suivantes :

  Compartiment source :

  ```
  "s3-outposts:GetObjectVersionForReplication",
  "s3-outposts:GetObjectVersionTagging"
  ```

  Compartiments de destination :

  ```
  "s3-outposts:ReplicateObject",
  "s3-outposts:ReplicateDelete",
  "s3-outposts:ReplicateTags"
  ```

- Amazon EventBridge peut vous avertir lorsque des objets ne se répliquent pas vers leurs Outposts de destination. Pour de plus amples informations, veuillez consulter Utilisation d'EventBridge pour la réplication S3 sur Outposts.

Utilisation d'EventBridge pour la réplication S3 sur Outposts

Amazon S3 sur Outposts est intégré à Amazon EventBridge et utilise l'espace de noms `s3-outposts`. EventBridge est un service de bus d'événements sans serveur que vous pouvez utiliser pour connecter vos applications à des données provenant de diverses sources. Pour plus

d'informations, consultez Qu'est-ce qu'Amazon EventBridge ? dans le Guide de l'utilisateur Amazon EventBridge.

Pour vous aider à résoudre les problèmes de configuration de la réplication, vous pouvez également configurer Amazon EventBridge afin qu'il reçoive des notifications sur les événements d'échec de la réplication. Les notifications d'EventBridge peuvent vous avertir dans les cas où les objets ne sont pas répliqués vers leurs Outposts de destination. Pour plus d'informations sur l'état actuel d'un objet répliqué, consultez Vue d'ensemble des statuts de réplication.

S3 sur Outposts peut envoyer des événements à EventBridge dès que certains événements se produisent dans votre compartiment Outposts. Contrairement à d'autres destinations, vous n'avez pas besoin de sélectionner les types d'événements que vous souhaitez proposer. Vous pouvez également utiliser les règles EventBridge pour acheminer des événements vers des cibles supplémentaires. Une fois EventBridge activé, S3 sur Outposts envoie tous les événements suivants à EventBridge.

Type d'événement	Description	Espace de noms
`Operation FailedReplication`	La réplication d'un objet au sein d'une règle de réplication a échoué. Pour plus d'informations sur les raisons de l'échec de la réplication S3 sur Outposts, consultez Utilisation d'EventBridge pour afficher les raisons de l'échec de la réplication S3 sur Outposts.	`s3-outposts`

Utilisation d'EventBridge pour afficher les raisons de l'échec de la réplication S3 sur Outposts

Le tableau suivant répertorie les raisons de l'échecs de la réplication S3 sur Outposts. Vous pouvez configurer une règle EventBridge pour publier et afficher la raison de l'échec via Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS), AWS Lambda ou Amazon CloudWatch Logs. Pour plus d'informations sur les autorisations requises pour utiliser ces ressources pour EventBridge, consultez Using resource-based policies for EventBridge (Utilisation de politiques basées sur les ressources pour EventBridge).

Raison de l'échec de la réplication	Description
`AssumeRoleNotPermitted`	S3 sur Outposts ne peut pas assumer le rôle AWS Identity and Access Management (IAM) spécifié dans la configuration de la réplicati on.
`DstBucketNotFound`	S3 sur Outposts n'est pas en mesure de trouver le compartiment de destination spécifié dans la configuration de la réplicati on.
`DstBucketUnversioned`	La gestion des versions n'est pas activée sur le compartiment de destination Outposts. Pour répliquer des objets avec la réplicati on S3 sur Outposts, vous devez activer la gestion des versions pour le compartiment de destination.
`DstDelObjNotPermitted`	S3 sur Outposts n'est pas en mesure de répliquer les suppressions vers le compartim ent de destination. L'autorisation `s3-outpos ts:ReplicateDelete` peut être manquante pour le compartiment de destinati on.
`DstMultipartCompleteNotPermitted`	S3 sur Outposts n'est pas en mesure de terminer le chargement partitionné des objets dans le compartiment de destination. L'autoris ation `s3-outposts:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstMultipartInitNotPermitted`	S3 sur Outposts n'est pas en mesure de lancer le chargement partitionné des objets vers le compartiment de destination. L'autoris ation `s3-outposts:ReplicateObject`

Raison de l'échec de la réplication	Description
	peut être manquante pour le compartiment de destination.
`DstMultipartPartUploadNotPermitted`	S3 sur Outposts n'est pas en mesure de charger le chargement partitionné des objets dans le compartiment de destination. L'autorisation `s3-outposts:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstOutOfCapacity`	S3 sur Outposts n'est pas en mesure de répliquer l'Outpost de destination car l'Outpost n'est pas compris dans la capacité de stockage de S3.
`DstPutObjNotPermitted`	S3 sur Outposts n'est pas en mesure de répliquer les objets vers le compartiment de destination. L'autorisation `s3-outposts:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstPutTaggingNotPermitted`	S3 sur Outposts n'est pas en mesure de répliquer les balises d'objets vers le compartiment de destination. L'autorisation `s3-outposts:ReplicateObject` peut être manquante pour le compartiment de destination.
`DstVersionNotFound`	S3 sur Outposts n'est pas en mesure de trouver la version d'objet requise dans le compartiment de destination pour répliquer les métadonnées de cette version d'objet.

Raison de l'échec de la réplication	Description
`SrcBucketReplicationConfigMissing`	S3 sur Outposts n'est pas en mesure de trouver une configuration de réplication pour le point d'accès associé au compartiment Outposts source.
`SrcGetObjNotPermitted`	S3 sur Outposts n'est pas en mesure d'accéder à l'objet dans le compartiment source pour la réplication. L'autorisation `s3-outposts:GetObjectVersionForReplication` peut être manquante pour le compartiment source.
`SrcGetTaggingNotPermitted`	S3 sur Outposts n'est pas en mesure d'accéder aux informations de balises d'objets dans le compartiment source. L'autorisation `s3-outposts:GetObjectVersionTagging` peut être manquante pour le compartiment source.
`SrcHeadObjectNotPermitted`	S3 sur Outposts n'est pas en mesure de récupérer les métadonnées d'objets du compartiment source. L'autorisation `s3-outposts:GetObjectVersionForReplication` peut être manquante pour le compartiment source.
`SrcObjectNotEligible`	L'objet n'est pas éligible à la réplication. L'objet ou ses balises d'objet ne correspondent pas à la configuration de réplication.

Pour plus d'informations sur le dépannage de la réplication, consultez les rubriques suivantes :

- Création d'un rôle IAM
- Résolution des problèmes de réplication

Surveillance d'EventBridge avec CloudWatch

Amazon EventBridge s'intègre à Amazon CloudWatch à des fins de surveillance. EventBridge envoie automatiquement des métriques à CloudWatch toutes les minutes. Ces mesures incluent le nombre d'événements auxquels correspond une règle et le nombre de fois qu'une cible est appelée par une règle. Quand une règle s'exécute dans EventBridge, toutes les cibles associées à la règle sont appelées. Vous pouvez surveiller votre comportement EventBridge via CloudWatch de la manière suivante.

- Vous pouvez surveiller les métriques EventBridge disponibles pour vos règles EventBridge à partir du tableau de bord CloudWatch. Vous pouvez ensuite utiliser les fonctions de CloudWatch, telles que les alarmes CloudWatch, pour définir des alarmes sur certaines métriques. Si ces métriques atteignent les valeurs de seuil personnalisées que vous avez spécifiées dans les alarmes, vous recevez des notifications et pouvez agir en conséquence.
- Vous pouvez définir Amazon CloudWatch Logs comme cible de votre règle EventBridge. EventBridge crée ensuite des flux de journaux et CloudWatch Logs stocke le texte des événements sous forme d'entrées de journal. Pour plus d'informations, consultez EventBridge et CloudWatch Logs.

Pour plus d'informations sur le débogage de la livraison d'événements et de l'archivage d'événements EventBridge, consultez les rubriques suivantes :

- Politique relative aux nouvelles tentatives d'événements et utilisation de files d'attente de lettres mortes
- Archivage d'événements EventBridge

Partage de S3 sur Outposts en utilisant AWS RAM

Amazon S3 on Outposts prend en charge le partage de la capacité S3 entre plusieurs comptes au sein d'une organisation en utilisant AWS Resource Access Manager () AWS RAM. Avec le partage de S3 on Outposts, vous pouvez autoriser d'autres utilisateurs à créer et gérer des compartiments, des points de terminaison et des points d'accès sur votre Outpost.

Cette rubrique explique comment AWS RAM partager S3 sur Outposts et les ressources associées avec un autre membre de votre Compte AWS AWS organisation.

Prérequis

- Le compte propriétaire Outpost dispose d'une organisation configurée dans AWS Organizations. Pour plus d'informations, consultez Création d'une organisation dans le AWS Organizations Guide de l'utilisateur.
- L'organisation inclut Compte AWS celle avec laquelle vous souhaitez partager votre capacité S3 on Outposts. Pour de plus amples informations, consultez Envoi d'invitations à des Comptes AWS dans le Guide de l'utilisateur AWS Organizations .
- Sélectionnez l'une des options suivantes que vous voulez partager. La deuxième ressource (Subnets (Sous-réseaux) ou Outposts) doit être sélectionnée pour que les points de terminaison soient également accessibles. Les points de terminaison sont une exigence liée à la mise en réseau afin d'accéder aux données stockées dans S3 on Outposts.

Option 1	Option 2
S3 on Outposts Permet à l'utilisateur de créer des compartiments sur vos Outposts et points d'accès et d'ajouter des objets à ces compartiments.	S3 on Outposts Permet à l'utilisateur de créer des compartiments sur vos Outposts et points d'accès et d'ajouter des objets à ces compartiments.
Sous-réseaux Permet à l'utilisateur d'utiliser votre cloud privé virtuel (VPC) et les points de terminaison associés à votre sous-réseau.	Outposts Permet à l'utilisateur de voir les diagrammes de capacité S3 et la page d'accueil de la console AWS Outposts . Permet également aux utilisateurs de créer des sous-réseaux sur des Outposts partagés et de créer des points de terminaison.

Procédure

1. Connectez-vous au en AWS Management Console utilisant le propriétaire de Compte AWS l'Outpost, puis ouvrez la AWS RAM console à l'adresse https://console.aws.amazon.com/ram.

2. Assurez-vous d'avoir activé le partage avec AWS Organizations in AWS RAM. Pour de plus amples informations, consultez Activer le partage des ressources dans AWS Organizations dans le Guide de l'utilisateur AWS RAM .
3. Utilisez l'option 1 ou l'option 2 dans les exigences pour créer un partage de ressources. Si vous disposez de plusieurs ressources S3 on Outposts, sélectionnez les Amazon Resource Names (ARN) des ressources que vous souhaitez partager. Pour activer les points de terminaison, partagez votre sous-réseau ou Outpost.

   Pour de plus amples informations sur la création d'un partage de ressources, consultez Créer un partage de ressources dans le AWS RAM Guide de l'utilisateur.
4. Les Compte AWS personnes avec lesquelles vous avez partagé vos ressources devraient désormais être en mesure d'utiliser S3 sur les Outposts. Selon l'option que vous avez sélectionnée dans les prérequis, fournissez les informations suivantes à l'utilisateur du compte :

Option 1	Option 2
L'ID Outpost	L'ID Outpost
L'ID de VPC	
L'ID de sous-réseau	
L'ID du groupe de sécurité	

> **Note**
>
> L'utilisateur peut confirmer que les ressources ont été partagées avec lui à l'aide de la AWS RAM console, de la AWS Command Line Interface (AWS CLI), AWS des SDK ou de l'API REST. L'utilisateur peut consulter ses partages de ressources existants à l'aide de la commande get-resource-sharesCLI.

Exemples d'utilisation

Une fois que vous avez partagé vos ressources S3 on Outposts avec un autre compte, ce compte peut gérer des compartiments et des objets sur votre Outpost. Si vous avez partagé la ressource Subnets (Sous-réseaux), ce compte peut alors utiliser le point de terminaison que vous avez créé.

Les exemples suivants montrent comment un utilisateur peut utiliser le AWS CLI pour interagir avec votre Outpost après avoir partagé ces ressources.

Example : Créer un compartiment

L'exemple suivant crée un bucket nommé *example-s3-bucket1* sur l'Outpost. *op-01ac5d28a6a232904* Avant d'utiliser cette commande, remplacez chaque *user input placeholder* par les valeurs appropriées pour votre cas d'utilisation.

```
aws s3control create-bucket --bucket example-s3-bucket1 --outpost-
id op-01ac5d28a6a232904
```

Pour de plus amples informations sur cette commande, veuillez consulter create-bucket dans le document AWS CLI Reference.

Example : Créer un point d'accès

L'exemple suivant crée un point d'accès sur un Outpost, à l'aide des exemples de paramètre du tableau suivant. Avant d'utiliser cette commande, remplacez ces *user input placeholder* valeurs et le Région AWS code par les valeurs appropriées à votre cas d'utilisation.

Paramètre	Valeur
ID de compte	*111122223333*
Nom du point d'accès	*example-outpost-access-point*
ID Outpost	*op-01ac5d28a6a232904*
Nom du compartiment Outpost	*example-s3-bucket1*
ID du VPC	*vpc-1a2b3c4d5e6f7g8h9*

> **Note**
>
> Le paramètre Account ID doit être l' Compte AWS ID du propriétaire du bucket, qui est l'utilisateur partagé.

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-access-point \
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-s3-bucket1 \
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

Pour plus d'informations sur cette commande, reportez-vous create-access-pointà la section AWS CLI Référence.

Example : Charger un objet

L'exemple suivant permet de charger le fichier *my_image.jpg* depuis le système de fichiers local de l'utilisateur vers un objet nommé *images/my_image.jpg* via le point d'accès *example-outpost-access-point* sur l'Outpost *op-01ac5d28a6a232904*, détenu par le compte AWS *111122223333*. Avant d'utiliser cette commande, remplacez ces *user input placeholder* valeurs et le Région AWS code par les valeurs appropriées à votre cas d'utilisation.

```
aws s3api put-object --bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point \
--body my_image.jpg --key images/my_image.jpg
```

Pour de plus amples informations sur cette commande, veuillez consulter put-object dans le document AWS CLI Reference.

> **Note**
>
> Si cette opération aboutit à une erreur de ressource introuvable ou ne répond pas, il se peut que votre VPC n'ait pas de point de terminaison partagé.
> Pour vérifier s'il existe un point de terminaison partagé, utilisez la list-shared-endpoints AWS CLI commande. S'il n'existe aucun point de terminaison partagé, demandez au propriétaire Outpost de vous en créer un. Pour plus d'informations, consultez ListSharedEndpointsle manuel Amazon Simple Storage Service API Reference.

Example : Créer un point de terminaison

L'exemple suivant d'utilisation permet de créer un point de terminaison sur un Outpost partagé. Avant d'utiliser cette commande, remplacez les valeurs *user input placeholder* pour l'ID Outpost, l'ID de sous-réseau et l'ID du groupe de sécurité par les valeurs appropriées pour votre cas d'utilisation.

> **Note**
>
> L'utilisateur ne peut effectuer cette opération que si le partage de ressources comprend la ressource Outposts.

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --
security-group-id XXXXXXX
```

Pour de plus amples informations sur cette commande, veuillez consulter create-endpoint dans le document AWS CLI Reference.

Autre Services AWS utilisant S3 on Outposts

D'autres Services AWS fonctionnant localement avec votre AWS Outposts peuvent également utiliser votre capacité Amazon S3 on Outposts. Dans Amazon CloudWatch, l'espace de noms `S3Outposts` affiche des métriques détaillées pour les compartiments de S3 on Outposts, mais ces métriques n'incluent pas l'utilisation pour les autres Services AWS. Pour gérer votre capacité S3 on Outposts qui est consommée par d'autres Services AWS, consultez les informations dans le tableau suivant.

Service AWS	Description	En savoir plus
Amazon S3	Toute utilisation directe de S3 on Outposts dispose d'un compte et d'un compartiment CloudWatch correspondant.	Afficher les métriques
Amazon Elastic Block Store (Amazon EBS)	Pour Amazon EBS on Outposts, vous pouvez sélection ner un Outpost AWS comme destination de l'instantané et le stocker localement dans votre S3 on Outpost.	En savoir plus
Amazon Relationa l Database Service (Amazon RDS)	Vous pouvez utiliser les sauvegardes locales Amazon RDS pour stocker vos sauvegardes RDS localement sur votre Outpost.	En savoir plus

Surveillance de S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos AWS Outposts et stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts fournit une nouvelle classe de stockage, S3 Outposts (OUTPOSTS), qui utilise les API Amazon S3 et est conçue pour stocker les données de manière durable et redondante sur plusieurs appareils et serveurs de votre entreprise. AWS Outposts Vous communiquez avec votre compartiment Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts via l'API AWS Management Console, AWS Command Line Interface (AWS CLI), les AWS SDK ou l'API REST. Pour plus d'informations, consultez Qu'est-ce que Amazon S3 sur Outposts ?.

Pour plus d'informations sur la surveillance de votre capacité de stockage Amazon S3 sur Outposts, consultez les rubriques suivantes.

Rubriques

- Gérer la capacité de S3 on Outposts avec Amazon Metrics CloudWatch
- Recevoir des notifications d'événements S3 on Outposts à l'aide d'Amazon Events CloudWatch
- Surveillance de S3 sur Outposts avec des journaux AWS CloudTrail

Gérer la capacité de S3 on Outposts avec Amazon Metrics CloudWatch

Pour vous aider à gérer la capacité S3 fixe de votre Outpost, nous vous recommandons de créer des CloudWatch alertes qui vous avertissent lorsque l'utilisation de votre stockage dépasse un certain seuil. Pour plus d'informations sur les CloudWatch métriques de S3 sur Outposts, consultez. CloudWatch métriques S'il n'y a pas assez d'espace pour stocker un objet sur votre Outpost, l'API renvoie une exemption de capacité insuffisante (ICE). Pour libérer de l'espace, vous pouvez créer des CloudWatch alarmes qui déclenchent la suppression explicite des données ou utiliser une politique d'expiration du cycle de vie pour faire expirer les objets. Pour enregistrer les données avant de les supprimer, vous pouvez AWS DataSync les utiliser pour copier les données de votre compartiment Amazon S3 on Outposts vers un compartiment S3 dans un. Région AWS Pour plus d'informations sur l'utilisation DataSync, voir Getting Started with AWS DataSync dans le guide de AWS DataSync l'utilisateur.

CloudWatch métriques

L'espace de noms `S3Outposts` inclut les métriques suivantes pour les compartiments Amazon S3 sur Outposts. Vous pouvez surveiller le nombre total d'octets S3 sur Outposts alloués, le nombre total d'octets libres disponibles pour les objets et la taille totale de tous les objets pour un compartiment donné. Des métriques liées aux compartiments ou aux comptes existent pour toute utilisation directe de S3. L'utilisation indirecte de S3, telle que le stockage d'instantanés locaux Amazon Elastic Block Store ou de sauvegardes Amazon Relational Database Service sur un Outpost, consomme de la capacité S3, mais n'est pas incluse dans les métriques liées aux compartiments ou aux comptes. Pour plus d'informations sur les instantanés locaux Amazon EBS, consultez Instantanés locaux Amazon EBS sur Outposts. Pour consulter votre rapport sur les coûts Amazon EBS, accédez à https://console.aws.amazon.com/billing/.

> **Note**
>
> S3 on Outposts ne prend en charge que les métriques suivantes et aucune autre métrique Amazon S3.
> S3 on Outposts ayant une limite de capacité fixe, nous vous recommandons de créer des CloudWatch alarmes pour vous avertir lorsque l'utilisation de votre stockage dépasse un certain seuil.

Métrique	Description	Période	Unités	Type
`Outpost` `talByte`	Capacité totale allouée en octets pour un Outpost	5 minutes	Octets	S3 on Outposts
`Outpostl` `eeBytes`	Nombre d'octets libres disponibles sur un Outpost pour stocker les données des clients.	5 minutes	Octets	S3 on Outposts
`BucketU` `dBytes`	Taille totale de tous les objets pour le compartiment donné.	5 minutes	Octets	S3 sur Outposts. Utilisation directe de S3 uniquement.
`Accountl` `edBytes`	Taille totale de tous les objets pour le compte Outposts spécifié.	5 minutes	Octets	S3 sur Outposts. Utilisation directe de S3 uniquement.

Métrique	Description	Période	Unités	Type
BytesPe ingRepl ation	Nombre total d'octets d'objets en attente de réplication pour une règle de réplication donnée. Pour plus d'informations sur l'activation des métriques de réplication, consultez la section Creating replication rules between Outposts (Création de règles de réplication entre Outposts).	5 minutes	Octets	Facultatif. Pour la réplication S3 sur Outposts.
Operati sPendin eplicat n	Nombre total d'opérations en attente de réplication pour une règle de réplication donnée. Pour plus d'informations sur l'activation des métriques de réplication, consultez la section Creating replication rules between Outposts (Création de règles de réplication entre Outposts).	5 minutes	Comptes	Facultatif. Pour la réplication S3 sur Outposts.
Replica onLaten	Nombre actuel de secondes de retard entre le compartiment de destination de réplication et le compartiment source pour une règle de réplication donnée. Pour plus d'informations sur l'activation des métriques de réplication, consultez la section Creating replication rules between Outposts (Création de règles de réplication entre Outposts).	5 minutes	Secondes	Facultatif. Pour la réplication S3 sur Outposts.

Recevoir des notifications d'événements S3 on Outposts à l'aide d'Amazon Events CloudWatch

Vous pouvez utiliser CloudWatch Events pour créer une règle pour tout événement d'API Amazon S3 on Outposts. Lorsque vous créez une règle, vous pouvez choisir d'être averti via toutes les CloudWatch cibles prises en charge, notamment Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) et. AWS Lambda Pour plus d'informations, consultez la liste des AWS services pouvant être la cible d' CloudWatch événements dans le guide de l'utilisateur Amazon CloudWatch Events. Pour choisir un service cible qui fonctionnera avec votre S3 sur Outposts, consultez la section Création d'une règle d' CloudWatch événements qui se déclenche lors d'un appel d' AWS API AWS CloudTrail dans le guide de l'utilisateur Amazon CloudWatch Events.

> **Note**
>
> Pour les opérations d'objets S3 on Outposts, les événements d'appel d' AWS API envoyés par S3 ne CloudTrail respecteront vos règles que si vous avez configuré des pistes (éventuellement avec des sélecteurs d'événements) pour recevoir ces événements. Pour plus d'informations, consultez la section Utilisation des fichiers CloudTrail journaux dans le Guide de AWS CloudTrail l'utilisateur.

Example

Voici un exemple de règle pour l'opération `DeleteObject`. Pour utiliser cet exemple de règle, remplacez *`example-s3-bucket1`* par le nom de votre compartiment S3 sur Outposts.

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
    "eventName": [
```

```
      "DeleteObject"
    ],
    "requestParameters": {
      "bucketName": [
        "example-s3-bucket1"
      ]
    }
  }
}
```

Surveillance de S3 sur Outposts avec des journaux AWS CloudTrail

Amazon S3 on Outposts est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou un utilisateur Service AWS dans S3 sur Outposts. Vous pouvez utiliser AWS CloudTrail pour obtenir des informations sur des demandes au niveau du compartiment et au niveau de l'objet sur S3 sur Outposts pour auditer et journaliser votre activité d'événements S3 sur Outposts. Pour activer les événements de CloudTrail données pour tous vos compartiments Outposts ou pour une liste de compartiments Outposts spécifiques, vous devez créer un suivi manuellement dans. CloudTrail Pour plus d'informations sur les entrées de fichiers CloudTrail journaux, consultez S3 sur les entrées de fichiers journaux d'Outposts.

> **Note**
>
> - Il est recommandé de créer une politique de cycle de vie pour votre bucket Outposts d'événements de AWS CloudTrail données. Configurez la politique de cycle de vie pour supprimer périodiquement les fichiers journaux après le délai à l'issue duquel vous devez les auditer. Cela permet de réduire la quantité de données analysées par Amazon Athena pour chaque requête. Pour plus d'informations, consultez Configuration du cycle de vie d'un bucket.
> - Pour des exemples expliquant comment interroger les CloudTrail journaux, consultez le billet de blog AWS consacré au Big Data Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail et Amazon Athena.

Activer la CloudTrail journalisation des objets dans un compartiment S3 on Outposts

Vous pouvez utiliser la console Amazon S3 pour configurer un AWS CloudTrail suivi afin de consigner les événements de données relatifs aux objets d'un compartiment Amazon S3 on

Outposts. CloudTrail prend en charge la journalisation de S3 sur les opérations d'API au niveau des objets d'Outposts, telles `GetObject` que, et. `DeleteObject` `PutObject` Ces événements sont des événements de données.

Par défaut, les CloudTrail sentiers n'enregistrent pas les événements liés aux données. Cependant, vous pouvez configurer des journaux de suivi pour journaliser des événements de données pour les compartiments S3 sur Outposts que vous spécifiez ou journaliser des événements de données pour tous les compartiments S3 sur Outposts dans votre Compte AWS. Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.

CloudTrail ne renseigne pas les événements de données dans l'historique des CloudTrail événements. De plus, les opérations d'API S3 on Outposts au niveau du bucket ne sont pas toutes renseignées dans l'historique des événements. CloudTrail Pour plus d'informations sur la manière d'interroger CloudTrail les journaux, consultez les sections Utilisation CloudWatch des modèles de filtre Amazon Logs et Amazon Athena pour interroger les CloudTrail journaux dans le AWS Knowledge Center.

Pour configurer un journal de suivi afin de journaliser les événements de données pour un compartiment S3 sur Outposts, vous pouvez utiliser la console AWS CloudTrail ou la console Amazon S3. Si vous configurez un journal pour consigner les événements de données pour tous les compartiments S3 on Outposts de vos compartiments Compte AWS, il est plus facile d'utiliser la console. CloudTrail Pour plus d'informations sur l'utilisation de la CloudTrail console pour configurer un journal des événements de données S3 on Outposts, consultez la section Événements liés aux données dans le guide de l'AWS CloudTrail utilisateur.

> **Important**
>
> Des frais supplémentaires s'appliquent pour les événements de données. Pour en savoir plus, consultez Tarification de AWS CloudTrail.

La procédure suivante explique comment utiliser la console Amazon S3 pour configurer un journal afin de CloudTrail consigner les événements de données d'un compartiment S3 on Outposts.

> **Note**
>
> La personne Compte AWS qui crée le compartiment en est propriétaire et est la seule à pouvoir configurer les événements de données S3 on Outposts à envoyer. AWS CloudTrail

Pour activer la journalisation des événements de CloudTrail données pour les objets d'un bucket S3 on Outposts

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Outposts buckets (Compartiments Outposts).
3. Choisissez le nom du bucket Outposts dont vous souhaitez enregistrer les événements de données en utilisant. CloudTrail
4. Choisissez Propriétés.
5. Dans la section AWS CloudTrail des événements de données, choisissez Configurer dans CloudTrail.

   La AWS CloudTrail console s'ouvre.

   Vous pouvez créer une nouvelle CloudTrail piste ou réutiliser une piste existante et configurer les événements de données S3 on Outposts pour qu'ils soient enregistrés dans votre trace.
6. Sur la page du tableau de bord de la CloudTrail console, choisissez Create trail.
7. Sur la page Étape 1 Choisir les attributs du journal de suivi, donnez un nom au journal de suivi, choisissez un compartiment S3 pour stocker les journaux de suivi, spécifiez les autres paramètres souhaités, puis cliquez sur Suivant.
8. Sur la page Étape 2 Choisir des événements de journaux, sous Type d'événement, choisissez Événements de données.

   Pour Type d'événement de données, choisissez S3 Outposts. Choisissez Suivant.

   > **Note**
   >
   > - Lorsque vous créez un journal de suivi et que vous configurez la journalisation des événements de données pour S3 sur Outposts, vous devez spécifier correctement le type d'événement de données.
   >   - Si vous utilisez la CloudTrail console, choisissez le type d'événement S3 Outposts for Data. Pour plus d'informations sur la création de pistes dans la CloudTrail console, consultez la section Création et mise à jour d'une piste avec la console dans le guide de AWS CloudTrail l'utilisateur. Pour plus d'informations sur la configuration de la journalisation des événements de données S3 on Outposts

dans la CloudTrail console, consultez la section Journalisation des événements de données pour les objets Amazon S3 dans le guide de l'AWS CloudTrail utilisateur.

- Si vous utilisez le AWS Command Line Interface (AWS CLI) ou les AWS SDK, définissez le `resources.type` champ sur. `AWS::S3Outposts::Object` Pour plus d'informations sur la façon de consigner les événements de données S3 on Outposts avec le AWS CLI, consultez la section Log S3 on Outposts dans le guide de l'utilisateur.AWS CloudTrail

- Si vous utilisez la CloudTrail console ou la console Amazon S3 pour configurer un journal des événements de données pour un compartiment S3 on Outposts, la console Amazon S3 indique que la journalisation au niveau des objets est activée pour le compartiment.

9. Sur la page Étape 3 Vérifier et créer, passez en revue les attributs du suivi et les événements du journal que vous avez configurés. Choisissez ensuite Créer un journal de suivi.

Pour désactiver la journalisation des événements de CloudTrail données pour les objets d'un bucket S3 on Outposts

1. Connectez-vous à la CloudTrail console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/cloudtrail/.
2. Dans le panneau de navigation de gauche, choisissez Journaux de suivi.
3. Choisissez le nom du journal de suivi que vous avez créé pour journaliser les événements de votre compartiment S3 sur Outposts.
4. Sur la page de détails de votre journal de suivi, choisissez Arrêter la journalisation dans le coin supérieur droit.
5. Dans la boîte de dialogue qui s'affiche, cliquez sur Arrêter la journalisation.

Développement avec Amazon S3 on Outposts

Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur votre AWS Outposts afin de stocker et récupérer facilement des objets sur site pour des applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. S3 on Outposts propose une nouvelle classe de stockage, appelée S3 Outposts (`OUTPOSTS`), qui utilise les API Amazon S3 et est conçue pour stocker de manière durable et redondante des données sur plusieurs appareils et serveurs sur vos AWS Outposts. Vous communiquez avec votre compartiment

Outpost à l'aide d'un point d'accès et d'une connexion de point de terminaison via un cloud privé virtuel (VPC). Vous pouvez utiliser les mêmes API et fonctions sur vos compartiments Outpost que sur les compartiments Amazon S3, telles que les stratégies d'accès, le chiffrement et le balisage. Vous pouvez utiliser S3 sur Outposts par le biais de la AWS Management Console, de la AWS CLI (AWS Command Line Interface), des kits SDK AWS ou d'une API REST. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon S3 sur Outposts ?.

Les rubriques suivantes fournissent des informations sur le développement avec S3 on Outposts.

Rubriques

- Opérations d'API Amazon S3 on Outposts
- Configurer les client de contrôle S3 pour S3 on Outposts à l'aide du kit SDK pour Java
- Envoyer des requêtes à S3 sur Outposts via IPv6

Opérations d'API Amazon S3 on Outposts

Cette rubrique répertorie les opérations d'API Amazon S3, Amazon S3 Control et Amazon S3 on Outposts que vous pouvez utiliser avec Amazon S3 on Outposts.

Rubriques

- Opérations d'API Amazon S3 pour la gestion des objets
- Opérations d'API de contrôle Amazon S3 pour la gestion des compartiments
- Opérations d'API S3 sur Outposts pour la gestion d'Outposts

Opérations d'API Amazon S3 pour la gestion des objets

S3 on Outposts est conçu pour utiliser les mêmes opérations d'API sur les objets qu'Amazon S3. Vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outpost. Lorsque vous utilisez une opération d'API d'objet avec S3 sur Outposts, vous fournissez l'Amazon Resource Name (ARN) du point d'accès Outposts ou l'alias de point d'accès. Pour plus d'informations sur les alias de point d'accès, consultez Utilisation d'un alias de type compartiment pour le point d'accès de votre compartiment S3 sur Outposts.

Amazon S3 on Outposts prend en charge les opérations suivantes d'API Amazon S3 :

- AbortMultipartUpload

- CompleteMultipartUpload
- CopyObject
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- DeleteObjectTagging
- GetObject
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- ListObjectVersions
- ListParts
- PutObject
- PutObjectTagging
- UploadPart
- UploadPartCopy

Opérations d'API de contrôle Amazon S3 pour la gestion des compartiments

S3 on Outposts prend en charge les opérations d'API de contrôle Amazon S3 suivantes pour une utilisation avec les compartiments.

- CreateAccessPoint
- CreateBucket
- DeleteAccessPoint
- DeleteAccessPointPolicy
- DeleteBucket

- DeleteBucketLifecycleConfiguration
- DeleteBucketPolicy
- DeleteBucketReplication
- DeleteBucketTagging
- GetAccessPoint
- GetAccessPointPolicy
- GetBucket
- GetBucketLifecycleConfiguration
- GetBucketPolicy
- GetBucketReplication
- GetBucketTagging
- GetBucketVersioning
- ListAccessPoints
- ListRegionalBuckets
- PutAccessPointPolicy
- PutBucketLifecycleConfiguration
- PutBucketPolicy
- PutBucketReplication
- PutBucketTagging
- PutBucketVersioning

Opérations d'API S3 sur Outposts pour la gestion d'Outposts

S3 on Outposts prend en charge les opérations d'API Amazon S3 on Outposts suivantes pour la gestion des points de terminaison.

- CreateEndpoint
- DeleteEndpoint
- ListEndpoints
- ListOutpostsWithS3
- ListSharedEndpoints

Configurer les client de contrôle S3 pour S3 on Outposts à l'aide du kit SDK pour Java

L'exemple suivant illustre la configuration du client de contrôle Amazon S3 pour S3 on Outposts à l'aide du AWS SDK for Java. Pour utiliser cet exemple, remplacez chaque *`user input placeholder`* par vos propres informations.

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSAccessKey;
    String secretKey = SecretAccessKey;
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

    return AWSS3ControlClient.builder().enableUseArnRegion()
            .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
            .build();

}
```

Envoyer des requêtes à S3 sur Outposts via IPv6

Les points de terminaison à double pile Amazon S3 on Outposts et S3 on Outposts prennent en charge les demandes adressées aux compartiments S3 on Outposts en utilisant le protocole IPv6 ou IPv4. Grâce à la prise en charge IPv6 de S3 on Outposts, vous pouvez accéder à vos buckets et aux ressources de votre plan de contrôle et les exploiter via les API S3 on Outposts sur des réseaux IPv6.

> **Note**
>
> Les actions d'objets S3 on Outposts (telles que `PutObject` ou`GetObject`) ne sont pas prises en charge sur les réseaux IPv6.

Il n'y a aucun frais supplémentaire pour accéder à S3 sur les Outposts via les réseaux IPv6. Pour plus d'informations sur S3 on Outposts, consultez la section Tarification de S3 on Outposts.

Rubriques

Mise en route avec IPv6

Pour envoyer une demande à un bucket S3 on Outposts via IPv6, vous devez utiliser un point de terminaison à double pile. La section suivante décrit comment envoyer des demandes via IPv6 à l'aide de points de terminaison Dual-Stack.

Les points suivants sont importants à prendre en compte avant d'essayer d'accéder à un bucket S3 on Outposts via IPv6 :

- Le client et le réseau accédant au compartiment doivent être autorisés à utiliser le protocole IPv6.
- Les demandes de type hébergement virtuel et type chemin sont prises en charge pour un accès via IPv6. Pour plus d'informations, consultez Utilisation de S3 sur les points de terminaison à double pile d'Outposts.
- Si vous utilisez le filtrage des adresses IP source dans votre utilisateur AWS Identity and Access Management (IAM) ou les politiques du bucket S3 on Outposts, vous devez mettre à jour les politiques pour inclure les plages d'adresses IPv6.

  > **Note**
  >
  > Cette exigence s'applique uniquement aux opérations du bucket S3 on Outposts et aux ressources du plan de contrôle sur les réseaux IPv6. Les actions d'objets Amazon S3 on Outposts ne sont pas prises en charge sur les réseaux IPv6.

- Lorsque vous utilisez le protocole IPv6, les fichiers journaux d'accès au serveur génèrent les adresses IP au format IPv6. Vous devez mettre à jour les outils, scripts et logiciels existants que vous utilisez pour analyser les fichiers journaux S3 on Outposts, afin qu'ils puissent analyser les adresses IP distantes au format IPv6. Les outils, scripts et logiciels mis à jour analyseront ensuite correctement les adresses IP distantes au format IPv6.

Utilisation de points de terminaison à double pile pour effectuer des demandes sur un réseau IPv6

Pour effectuer des requêtes avec les appels d'API S3 on Outposts via IPv6, vous pouvez utiliser des points de terminaison à double pile via ou un SDK. AWS CLI AWS Les opérations de l'API de contrôle Amazon S3 et les opérations de l'API S3 on Outposts fonctionnent de la même manière, que vous accédiez à S3 on Outposts via un protocole IPv6 ou un protocole IPv4. Sachez toutefois que les actions d'objets S3 on Outposts (telles que `PutObject` ou`GetObject`) ne sont pas prises en charge sur les réseaux IPv6.

Lorsque vous utilisez l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS, vous pouvez utiliser un paramètre ou un indicateur pour passer à un point de terminaison Dual-Stack. Vous pouvez également spécifier le point de terminaison à double pile directement en remplacement du point de terminaison S3 on Outposts dans le fichier de configuration.

Vous pouvez utiliser un point de terminaison à double pile pour accéder à un bucket S3 on Outposts via IPv6 à partir de l'une des options suivantes :

- La AWS CLI (consultez Utilisation de points de terminaison Dual-Stack avec l AWS CLI).
- Les kits SDK AWS (consultez Utilisation de S3 sur les points de terminaison à double pile d'Outposts à partir des SDK AWS).

Utilisation d'adresses IPv6 dans les stratégies IAM

Avant d'essayer d'accéder à un bucket S3 on Outposts à l'aide d'un protocole IPv6, assurez-vous que les politiques de bucket des utilisateurs IAM ou S3 on Outposts utilisées pour le filtrage des adresses IP sont mises à jour pour inclure les plages d'adresses IPv6. Si les politiques de filtrage des adresses IP ne sont pas mises à jour pour gérer les adresses IPv6, vous risquez de perdre l'accès à un bucket S3 on Outposts lorsque vous essayez d'utiliser le protocole IPv6.

Les politiques IAM qui filtrent les adresses IP utilisent des opérateurs de condition d'adresse IP. La politique de compartiment S3 on Outposts suivante identifie la plage d'adresses IP 54.240.143.* des adresses IPv4 autorisées à l'aide d'opérateurs de condition d'adresse IP. Toutes les adresses IP situées en dehors de cette plage se verront refuser l'accès au compartiment S3 on Outposts ()`DOC-EXAMPLE-BUCKET`. Comme toutes les adresses IPv6 se trouvent hors de la plage autorisée, cette stratégie empêche les adresses IPv6 d'accéder à `DOC-EXAMPLE-BUCKET`.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3outposts:*",
      "Resource": "arn:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/DOC-EXAMPLE-BUCKET/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Vous pouvez modifier l'`Condition`élément de la politique de compartiment S3 on Outposts pour autoriser les plages d'adresses IPv4 (`54.240.143.0/24`) et IPv6 (`2001:DB8:1234:5678::/64`), comme indiqué dans l'exemple suivant. Vous pouvez utiliser le même type de bloc `Condition` que celui indiqué dans l'exemple pour mettre à jour vos stratégies de compartiment et d'utilisateur IAM.

```
       "Condition": {
         "IpAddress": {
            "aws:SourceIp": [
              "54.240.143.0/24",
               "2001:DB8:1234:5678::/64"
             ]
          }
        }
```

Avant d'utiliser le protocole IPv6, vous devez mettre à jour toutes les stratégies de compartiment et d'utilisateur IAM pertinentes qui utilisent le filtrage des adresses IP afin d'autoriser les plages d'adresses IPv6. Nous vous recommandons de mettre à jour vos stratégies IAM avec les plages d'adresses IPv6 de votre organisation, en plus des plages IPv4 existantes. Pour obtenir un exemple de stratégie de compartiment autorisant l'accès à la fois via IPv6 etIPv4, veuillez consulter Restriction de l'accès à des adresses IP spécifiques.

Vous pouvez consulter vos stratégies utilisateur IAM à l'aide de la console IAM à l'adresse https://console.aws.amazon.com/iam/. Pour de plus amples informations sur IAM, veuillez consulter le Guide de l'utilisateur IAM. Pour plus d'informations sur la modification des politiques du bucket S3

on Outposts, consultez. Ajout ou modification d'une politique de compartiment pour un compartiment Amazon S3 on Outposts.

Test de compatibilité d'adresses IP

Si vous utilisez une instance Linux ou Unix, ou une plateforme macOS X, vous pouvez tester votre accès à un point de terminaison à double pile via IPv6. Par exemple, pour tester la connexion à Amazon S3 sur les points de terminaison Outposts via IPv6, utilisez la commande : `dig`

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

Si votre point de terminaison à double pile sur un réseau IPv6 est correctement configuré, la `dig` commande renvoie les adresses IPv6 connectées. Par exemple :

```
dig s3-outposts.us-west-2.api.aws AAAA +short

2600:1f14:2588:4800:b3a9:1460:159f:ebce

2600:1f14:2588:4802:6df6:c1fd:ef8a:fc76

2600:1f14:2588:4801:d802:8ccf:4e04:817
```

Utilisation d'IPv6 avec AWS PrivateLink

S3 on Outposts prend en charge le protocole IPv6 pour les AWS PrivateLink services et les points de terminaison. Grâce à la prise en AWS PrivateLink charge du protocole IPv6, vous pouvez vous connecter aux points de terminaison de service au sein de votre VPC via des réseaux IPv6, à partir de connexions sur site ou d'autres connexions privées. La prise en charge IPv6 AWS PrivateLinkde S3 on Outposts vous permet également d'intégrer des points de terminaison AWS PrivateLink à double pile. Pour savoir comment activer IPv6 pourAWS PrivateLink, voir Accélérer l'adoption d'IPv6 avec les AWS PrivateLink services et les points de terminaison.

> **Note**
>
> Pour mettre à jour le type d'adresse IP pris en charge d'IPv4 à IPv6, voir Modifier le type d'adresse IP pris en charge dans le Guide de l'AWS PrivateLinkutilisateur.

Utilisation d'IPv6 avec AWS PrivateLink

Si vous utilisez AWS PrivateLink IPv6, vous devez créer un point de terminaison d'interface VPC IPv6 ou à double pile. Pour connaître les étapes générales de création d'un point de terminaison VPC à l'aide duAWS Management Console, consultez la section Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface dans le guide de l'utilisateur. AWS PrivateLink

AWS Management Console

Utilisez la procédure suivante pour créer un point de terminaison VPC d'interface qui se connecte à S3 sur Outposts.

1. Connectez-vous à la AWS Management Console et ouvrez la console VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le panneau de navigation, choisissez Points de terminaison.
3. Choisissez Créer un point de terminaison.
4. Pour Service category (Catégorie de service), choisissez Services AWS.
5. Dans Nom du service, choisissez le service S3 on Outposts (com.amazonaws.us-east-1.s3-outposts).
6. Pour le VPC, choisissez le VPC à partir duquel vous allez accéder à S3 sur Outposts.
7. Pour les sous-réseaux, choisissez un sous-réseau par zone de disponibilité à partir duquel vous accéderez à S3 sur Outposts. Il n'est pas possible de sélectionner plusieurs sous-réseaux dans la même zone de disponibilité. Pour chaque sous-réseau que vous sélectionnez, une nouvelle interface réseau de point de terminaison est créée. Par défaut, les adresses IP des plages d'adresses IP des sous-réseaux sont attribuées aux interfaces réseau des points de terminaison. Pour désigner une adresse IP pour une interface réseau de point de terminaison, choisissez Designate IP addresses et entrez une adresse IPv6 dans la plage d'adresses de sous-réseau.
8. Pour le type d'adresse IP, choisissez Dualstack. Attribuez des adresses IPv4 et IPv6 aux interfaces réseau de vos terminaux. Cette option n'est prise en charge que si tous les sous-réseaux sélectionnés possèdent des plages d'adresses IPv4 et IPv6.
9. Pour les groupes de sécurité, choisissez les groupes de sécurité à associer aux interfaces réseau du point de terminaison pour le point de terminaison VPC. Par défaut, le groupe de sécurité par défaut est associé au VPC.
10. Pour Politique, choisissez Accès complet afin d’autoriser toutes les opérations de tous les principaux sur toutes les ressources via le point de terminaison de VPC. Sinon, choisissez

Personnalisé pour associer une politique de point de terminaison VPC qui contrôle les autorisations dont disposent les principaux pour effectuer des actions sur les ressources via le point de terminaison VPC. Cette option n'est disponible que si le service prend en charge les politiques de points de terminaison de VPC. Pour plus d'informations, consultez la section Politiques relatives aux terminaux.

11. (Facultatif) Pour ajouter une identification, choisissez Add new tag (Ajouter une identification) et saisissez la clé et la valeur de l'identification.
12. Choisissez Créer un point de terminaison.

Example — Politique relative aux compartiments S3 on Outposts

Pour permettre à S3 on Outposts d'interagir avec vos points de terminaison VPC, vous pouvez ensuite mettre à jour votre politique S3 on Outposts comme suit :

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3-outposts:*",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

AWS CLI

> **Note**
>
> Pour activer le réseau IPv6 sur votre point de terminaison VPC, vous devez avoir `IPv6` défini le `SupportedIpAddressType` filtre pour S3 sur Outposts.

L'exemple suivant utilise la `create-vpc-endpoint` commande pour créer un nouveau point de terminaison d'interface à double pile.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpc-12345678 \
```

```
--vpc-endpoint-type Interface \
--service-name com.amazonaws.us-east-1.s3-outposts \
--subnet-id subnet-12345678 \
--security-group-id sg-12345678 \
--ip-address-type dualstack \
--dns-options "DnsRecordIpType=dualstack"
```

Selon la configuration du AWS PrivateLink service, les connexions de point de terminaison nouvellement créées devront peut-être être acceptées par le fournisseur de services de point de terminaison VPC avant de pouvoir être utilisées. Pour plus d'informations, consultez la section Accepter et rejeter les demandes de connexion des terminaux dans le Guide de AWS PrivateLink l'utilisateur.

L'exemple suivant utilise la `modify-vpc-endpoint` commande pour mettre à jour le point de terminaison VPC IPV uniquement vers un point de terminaison à double pile. Le point de terminaison à double pile permet d'accéder aux réseaux IPv4 et IPv6.

```
aws ec2 modify-vpc-endpoint \
--vpc-endpoint-id vpce-12345678 \
--add-subnet-ids subnet-12345678 \
--remove-subnet-ids subnet-12345678 \
--ip-address-type dualstack \
--dns-options "DnsRecordIpType=dualstack"
```

Pour plus d'informations sur la façon d'activer le réseau IPv6 pourAWS PrivateLink, voir Accélérer l'adoption d'IPv6 avec les AWS PrivateLink services et les points de terminaison.

Utilisation de S3 sur les points de terminaison à double pile d'Outposts

Les points de terminaison à double pile S3 on Outposts prennent en charge les requêtes adressées aux buckets S3 on Outposts via IPv6 et IPv4. Cette section décrit comment utiliser S3 sur les points de terminaison à double pile d'Outposts.

Rubriques

- Points de terminaison à double pile S3 on Outposts
- Utilisation de points de terminaison Dual-Stack avec l AWS CLI
- Utilisation de S3 sur les points de terminaison à double pile d'Outposts à partir des SDK AWS

Points de terminaison à double pile S3 on Outposts

Lorsque vous envoyez une demande à un point de terminaison à double pile, l'URL du compartiment S3 on Outposts est convertie en adresse IPv6 ou IPv4. Pour plus d'informations sur l'accès à un bucket S3 on Outposts via IPv6, consultez. Envoyer des requêtes à S3 sur Outposts via IPv6

Pour accéder à un bucket S3 on Outposts via un point de terminaison à double pile, utilisez un nom de point de terminaison de type chemin. S3 on Outposts ne prend en charge que les noms de point de terminaison régionaux à double pile, ce qui signifie que vous devez spécifier la région dans le nom.

Pour un point de terminaison FiPS de type chemin à double pile, utilisez la convention de dénomination suivante :

```
s3-outposts-fips.region.api.aws
```

Pour un point de terminaison non FIPS à double pile, utilisez la convention de dénomination suivante :

```
s3-outposts.region.api.aws
```

> **Note**
>
> Les noms de point de terminaison de type hébergé virtuel ne sont pas pris en charge dans S3 sur Outposts.

Utilisation de points de terminaison Dual-Stack avec l AWS CLI

Cette section fournit des exemples de commandes d'AWS CLI permettant d'envoyer des demandes à un point de terminaison Dual-Stack. Pour savoir comment configurer l'AWS CLI, consultez Commencer à utiliser le SDK AWS CLI and pour Java.

Vous définissez la valeur de configuration `use_dualstack_endpoint` sur `true` dans un profil de votre AWS Config fichier pour diriger toutes les demandes Amazon S3 effectuées par les `s3api` AWS CLI commandes `s3` and vers le point de terminaison à double pile pour la région spécifiée. Vous spécifiez la région dans le fichier de configuration ou dans une commande à l'aide de l'`--region`option.

Lorsque vous utilisez des points de terminaison à double pile avec le style d'adressageAWS CLI, seul le style d'`path`adressage est pris en charge. Le style d'adressage, défini dans le fichier de configuration, détermine si le nom du bucket figure dans le nom d'hôte ou dans l'URL. Pour plus d'informations, consultez s3outposts dans le Guide de l'utilisateur AWS CLI.

Pour utiliser un point de terminaison à double pile via leAWS CLI, utilisez le `--endpoint-url` paramètre avec le point de `https://s3-outposts-fips.`*`region`*`.api.aws` terminaison `http://s3.dualstack.`*`region`*`.amazonaws.com` ou pour toute `s3outposts` commande `s3control` or.

Par exemple :

```
$  aws s3control list-regional-buckets --endpoint-url https://s3-
outposts.region.api.aws
```

Utilisation de S3 sur les points de terminaison à double pile d'Outposts à partir des SDK AWS

Cette section fournit des exemples d'accès à un point de terminaison Dual-Stack à l'aide de kits SDK AWS.

AWS SDK for Java 2.xExemple de point de terminaison Dual-Stack (double pile) avec le kit

Les exemples suivants montrent comment utiliser les `S3OutpostsClient` classes `S3ControlClient` et pour activer les points de terminaison à double pile lors de la création d'un client S3 on Outposts à l'aide du. AWS SDK for Java 2.x Pour obtenir des instructions sur la création et le test d'un exemple Java fonctionnel pour Amazon S3 on Outposts, consultez. Commencer à utiliser le SDK AWS CLI and pour Java

Example — Créez une **`S3ControlClient`** classe avec des points de terminaison à double pile activés

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsRequest;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsResponse;
import software.amazon.awssdk.services.s3control.model.S3ControlException;


public class DualStackEndpointsExample1 {
```

```
    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");
        String accountId = "111122223333";
        String navyId = "9876543210";

        try {
            // Create an S3ControlClient with dual-stack endpoints enabled.
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                                                             .region(clientRegion)
                                                             .dualstackEnabled(true)
                                                             .build();
            ListRegionalBucketsRequest listRegionalBucketsRequest =
 ListRegionalBucketsRequest.builder()

       .accountId(accountId)

       .outpostId(navyId)

       .build();

            ListRegionalBucketsResponse listBuckets =
 s3ControlClient.listRegionalBuckets(listRegionalBucketsRequest);
            System.out.printf("ListRegionalBuckets Response: %s%n",
 listBuckets.toString());
       } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
       }
       catch (S3ControlException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
       } catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
 client
            // couldn't parse the response from Amazon S3 on Outposts.
            e.printStackTrace();
       }
    }
}
```

Example — Créez un **`S3OutpostsClient`** avec des points de terminaison à double pile activés

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3outposts.S3OutpostsClient;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsRequest;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsResponse;
import software.amazon.awssdk.services.s3outposts.model.S3OutpostsException;


public class DualStackEndpointsExample2 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");

        try {
            // Create an S3OutpostsClient with dual-stack endpoints enabled.
            S3OutpostsClient s3OutpostsClient = S3OutpostsClient.builder()
                                                                .region(clientRegion)
                                                                .dualstackEnabled(true)
                                                                .build();
            ListEndpointsRequest listEndpointsRequest =
 ListEndpointsRequest.builder().build();

            ListEndpointsResponse listEndpoints =
 s3OutpostsClient.listEndpoints(listEndpointsRequest);
            System.out.printf("ListEndpoints Response: %s%n",
 listEndpoints.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch (S3OutpostsException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
 client
            // couldn't parse the response from Amazon S3 on Outposts.
            e.printStackTrace();
        }
```

```
    }
}
```

Si vous utilisez AWS SDK for Java 2.x le sous Windows, vous devrez peut-être définir la propriété de machine virtuelle Java (JVM) suivante :

```
java.net.preferIPv6Addresses=true
```

Exemples de code pour Amazon S3 à l'aide de AWS kits SDK

Les exemples de code suivants montrent comment utiliser Amazon S3 avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés et dans des exemples interservices.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir une tâche spécifique en appelant plusieurs fonctions au sein d'un même service.

Les Exemples de services croisés sont des exemples d'applications fonctionnant sur plusieurs Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Mise en route

Hello Amazon S3

Les exemples de code suivants montrent comment démarrer avec Amazon S3.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Code pour le MakeLists fichier CMake C.txt.

```
# Set the minimum required version of CMake for this project.
```

```
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS s3)

# Set this project's name.
project("hello_s3")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
    # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

    # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
 may need to uncomment this
    # and set the proper subdirectory to the executables' location.

    AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_s3.cpp)

target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Code pour le fichier source hello_s3.cpp.

```
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <iostream>
#include <aws/core/auth/AWSCredentialsProviderChain.h>
using namespace Aws;
using namespace Aws::Auth;

/*
 *  A "Hello S3" starter application which initializes an Amazon Simple Storage
 Service (Amazon S3) client
 *  and lists the Amazon S3 buckets in the selected region.
 *
 *  main function
 *
 *  Usage: 'hello_s3'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        // You don't normally have to test that you are authenticated. But the
 S3 service permits anonymous requests, thus the s3Client will return "success"
 and 0 buckets even if you are unauthenticated, which can be confusing to a new
 user.
        auto provider =
 Aws::MakeShared<DefaultAWSCredentialsProviderChain>("alloc-tag");
        auto creds = provider->GetAWSCredentials();
        if (creds.IsEmpty()) {
            std::cerr << "Failed authentication" << std::endl;
        }

        Aws::S3::S3Client s3Client(clientConfig);
        auto outcome = s3Client.ListBuckets();
```

```
        if (!outcome.IsSuccess()) {
            std::cerr << "Failed with error: " << outcome.GetError() <<
 std::endl;
            result = 1;
        } else {
            std::cout << "Found " << outcome.GetResult().GetBuckets().size()
                      << " buckets\n";
            for (auto &bucket: outcome.GetResult().GetBuckets()) {
                std::cout << bucket.GetName() << std::endl;
            }
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for C++ API.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/s3"
```

```
)

// main uses the AWS SDK for Go V2 to create an Amazon Simple Storage Service
// (Amazon S3) client and list up to 10 buckets in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 s3Client := s3.NewFromConfig(sdkConfig)
 count := 10
 fmt.Printf("Let's list up to %v buckets for your account.\n", count)
 result, err := s3Client.ListBuckets(context.TODO(), &s3.ListBucketsInput{})
 if err != nil {
  fmt.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
  return
 }
 if len(result.Buckets) == 0 {
  fmt.Println("You don't have any buckets!")
 } else {
  if count > len(result.Buckets) {
   count = len(result.Buckets)
  }
  for _, bucket := range result.Buckets[:count] {
   fmt.Printf("\t%v\n", *bucket.Name)
  }
 }
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloS3 {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        listBuckets(s3);
    }

    public static void listBuckets(S3Client s3) {
        try {
            ListBucketsResponse response = s3.listBuckets();
            List<Bucket> bucketList = response.buckets();
            bucketList.forEach(bucket -> {
                System.out.println("Bucket Name: " + bucket.name());
```

```
            });

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";

// When no region or credentials are provided, the SDK will use the
// region and credentials from the local AWS config.
const client = new S3Client({});

export const helloS3 = async () => {
  const command = new ListBucketsCommand({});

  const { Buckets } = await client.send(command);
  console.log("Buckets: ");
  console.log(Buckets.map((bucket) => bucket.Name).join("\n"));
  return Buckets;
};
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for JavaScript API.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
use Aws\S3\S3Client;

$client = new S3Client(['region' => 'us-west-2']);
$results = $client->listBuckets();
var_dump($results);
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for PHP API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import boto3


def hello_s3():
    """
    Use the AWS SDK for Python (Boto3) to create an Amazon Simple Storage Service
    (Amazon S3) resource and list the buckets in your account.
    This example uses the default settings specified in your shared credentials
    and config files.
    """
```

```
    s3_resource = boto3.resource("s3")
    print("Hello, Amazon S3! Let's list your buckets:")
    for bucket in s3_resource.buckets.all():
        print(f"\t{bucket.name}")


if __name__ == "__main__":
    hello_s3()
```

- Pour plus de détails sur l'API, consultez ListBucketsle AWS manuel de référence de l'API SDK for Python (Boto3).

Exemples de code

- Actions pour Amazon S3 à l'aide de AWS kits SDK
  - Utilisation AbortMultipartUpload avec un AWS SDK ou une CLI
  - Utilisation AbortMultipartUploads avec un AWS SDK ou une CLI
  - Utilisation CompleteMultipartUpload avec un AWS SDK ou une CLI
  - Utilisation CopyObject avec un AWS SDK ou une CLI
  - Utilisation CreateBucket avec un AWS SDK ou une CLI
  - Utilisation CreateMultiRegionAccessPoint avec un AWS SDK ou une CLI
  - Utilisation CreateMultipartUpload avec un AWS SDK ou une CLI
  - Utilisation DeleteBucket avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketAnalyticsConfiguration avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketCors avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketEncryption avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketInventoryConfiguration avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketLifecycle avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketMetricsConfiguration avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketPolicy avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketReplication avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketTagging avec un AWS SDK ou une CLI
  - Utilisation DeleteBucketWebsite avec un AWS SDK ou une CLI
  - Utilisation DeleteObject avec un AWS SDK ou une CLI

- Utilisation DeleteObjectTagging avec un AWS SDK ou une CLI
- Utilisation DeleteObjects avec un AWS SDK ou une CLI
- Utilisation DeletePublicAccessBlock avec un AWS SDK ou une CLI
- Utilisation GetBucketAccelerateConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketAcl avec un AWS SDK ou une CLI
- Utilisation GetBucketAnalyticsConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketCors avec un AWS SDK ou une CLI
- Utilisation GetBucketEncryption avec un AWS SDK ou une CLI
- Utilisation GetBucketInventoryConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketLifecycleConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketLocation avec un AWS SDK ou une CLI
- Utilisation GetBucketLogging avec un AWS SDK ou une CLI
- Utilisation GetBucketMetricsConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketNotification avec un AWS SDK ou une CLI
- Utilisation GetBucketPolicy avec un AWS SDK ou une CLI
- Utilisation GetBucketPolicyStatus avec un AWS SDK ou une CLI
- Utilisation GetBucketReplication avec un AWS SDK ou une CLI
- Utilisation GetBucketRequestPayment avec un AWS SDK ou une CLI
- Utilisation GetBucketTagging avec un AWS SDK ou une CLI
- Utilisation GetBucketVersioning avec un AWS SDK ou une CLI
- Utilisation GetBucketWebsite avec un AWS SDK ou une CLI
- Utilisation GetObject avec un AWS SDK ou une CLI
- Utilisation GetObjectAcl avec un AWS SDK ou une CLI
- Utilisation GetObjectLegalHold avec un AWS SDK ou une CLI
- Utilisation GetObjectLockConfiguration avec un AWS SDK ou une CLI
- Utilisation GetObjectRetention avec un AWS SDK ou une CLI
- Utilisation GetObjectTagging avec un AWS SDK ou une CLI
- Utilisation GetPublicAccessBlock avec un AWS SDK ou une CLI
- Utilisation HeadBucket avec un AWS SDK ou une CLI
- Utilisation HeadObject avec un AWS SDK ou une CLI

- Utilisation ListBucketAnalyticsConfigurations avec un AWS SDK ou une CLI
- Utilisation ListBucketInventoryConfigurations avec un AWS SDK ou une CLI
- Utilisation ListBuckets avec un AWS SDK ou une CLI
- Utilisation ListMultipartUploads avec un AWS SDK ou une CLI
- Utilisation ListObjectVersions avec un AWS SDK ou une CLI
- Utilisation ListObjects avec un AWS SDK ou une CLI
- Utilisation ListObjectsV2 avec un AWS SDK ou une CLI
- Utilisation PutBucketAccelerateConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketAcl avec un AWS SDK ou une CLI
- Utilisation PutBucketCors avec un AWS SDK ou une CLI
- Utilisation PutBucketEncryption avec un AWS SDK ou une CLI
- Utilisation PutBucketLifecycleConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketLogging avec un AWS SDK ou une CLI
- Utilisation PutBucketNotification avec un AWS SDK ou une CLI
- Utilisation PutBucketNotificationConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketPolicy avec un AWS SDK ou une CLI
- Utilisation PutBucketReplication avec un AWS SDK ou une CLI
- Utilisation PutBucketRequestPayment avec un AWS SDK ou une CLI
- Utilisation PutBucketTagging avec un AWS SDK ou une CLI
- Utilisation PutBucketVersioning avec un AWS SDK ou une CLI
- Utilisation PutBucketWebsite avec un AWS SDK ou une CLI
- Utilisation PutObject avec un AWS SDK ou une CLI
- Utilisation PutObjectAcl avec un AWS SDK ou une CLI
- Utilisation PutObjectLegalHold avec un AWS SDK ou une CLI
- Utilisation PutObjectLockConfiguration avec un AWS SDK ou une CLI
- Utilisation PutObjectRetention avec un AWS SDK ou une CLI
- Utilisation RestoreObject avec un AWS SDK ou une CLI
- Utilisation SelectObjectContent avec un AWS SDK ou une CLI
- Utilisation UploadPart avec un AWS SDK ou une CLI

- Scénarios pour Amazon S3 utilisant des AWS SDK

- Création d'une URL présignée pour Amazon S3 à l'aide d'un SDK AWS
- Page Web répertoriant les objets Amazon S3 à l'aide d'un AWS SDK
- Supprimer les téléchargements partitionnés incomplets vers Amazon S3 à l'aide d'un SDK AWS
- Téléchargez tous les objets d'un compartiment Amazon Simple Storage Service (Amazon S3) dans un répertoire local.
- Obtenez un objet Amazon S3 à partir d'un point d'accès multirégional à l'aide d'un SDK AWS
- Obtenir un objet depuis un compartiment Amazon S3 à l'aide d'un AWS SDK, en spécifiant un en-tête If-Modified-Since
- Commencez à utiliser les buckets et les objets Amazon S3 à l'aide d'un SDK AWS
- Commencez à chiffrer les objets Amazon S3 à l'aide d'un AWS SDK
- Commencez à utiliser les balises pour les objets Amazon S3 à l'aide d'un AWS SDK
- Obtenez la configuration de conservation légale d'un objet Amazon S3 à l'aide d'un AWS SDK
- Utilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK
- Gérez les listes de contrôle d'accès (ACL) pour les compartiments Amazon S3 à l'aide d'un SDK AWS
- Gérez des objets Amazon S3 versionnés par lots à l'aide d'une fonction Lambda à l'aide d'un SDK AWS
- Analyser les URI Amazon S3 à l'aide d'un SDK AWS
- Réaliser une copie en plusieurs parties d'un objet Amazon S3 à l'aide d'un SDK AWS
- Effectuer un téléchargement partitionné d'un objet Amazon S3 à l'aide d'un SDK AWS
- Suivez le chargement ou le téléchargement d'un objet Amazon S3 à l'aide d'un AWS SDK
- Exemples d'approches pour les tests unitaires et d'intégration avec un AWS SDK
- Charger récursivement un répertoire local dans un compartiment Amazon Simple Storage Service (Amazon S3)
- Chargez ou téléchargez des fichiers volumineux vers et depuis Amazon S3 à l'aide d'un AWS SDK
- Chargez un flux de taille inconnue vers un objet Amazon S3 à l'aide d'un AWS SDK
- Utilisez des checksums pour travailler avec un objet Amazon S3 à l'aide d'un SDK AWS
- Travaillez avec des objets versionnés Amazon S3 à l'aide d'un SDK AWS

- Exemples de solutions sans serveur pour Amazon S3 utilisant des SDK AWS
  - Invoquer une fonction lambda à partir d'un déclencheur Amazon S3

- Exemples multiservices pour Amazon S3 utilisant AWS des kits de développement logiciel
  - Créer une application Amazon Transcribe
  - Convertissez du texte en parole et de nouveau en texte à l'aide d'un AWS SDK
  - Création d'une application de gestion des ressources photographiques permettant aux utilisateurs de gérer les photos à l'aide d'étiquettes
  - Créer une application Amazon Textract Explorer
  - Détectez le PPE dans les images avec Amazon Rekognition à l'aide d'un SDK AWS
  - Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK
  - Détecter les visages dans une image à l'aide d'un AWS SDK
  - Détectez des objets dans des images avec Amazon Rekognition à l'aide d'un SDK AWS
  - Détectez les personnes et les objets dans une vidéo avec Amazon Rekognition à l'aide d'un SDK AWS
  - Enregistrez les informations EXIF et autres informations sur les images à l'aide d'un SDK AWS
  - Transformez les données de votre application avec S3 Object Lambda

Actions pour Amazon S3 à l'aide de AWS kits SDK

Les exemples de code suivants montrent comment effectuer des actions Amazon S3 individuelles à l'aide des AWS SDK. Ces extraits appellent l'API Amazon S3 et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour une liste complète, consultez la Référence de l'API Amazon Simple Storage Service (Amazon S3).

Exemples

- Utilisation AbortMultipartUpload avec un AWS SDK ou une CLI
- Utilisation AbortMultipartUploads avec un AWS SDK ou une CLI
- Utilisation CompleteMultipartUpload avec un AWS SDK ou une CLI
- Utilisation CopyObject avec un AWS SDK ou une CLI
- Utilisation CreateBucket avec un AWS SDK ou une CLI
- Utilisation CreateMultiRegionAccessPoint avec un AWS SDK ou une CLI
- Utilisation CreateMultipartUpload avec un AWS SDK ou une CLI

- Utilisation DeleteBucket avec un AWS SDK ou une CLI
- Utilisation DeleteBucketAnalyticsConfiguration avec un AWS SDK ou une CLI
- Utilisation DeleteBucketCors avec un AWS SDK ou une CLI
- Utilisation DeleteBucketEncryption avec un AWS SDK ou une CLI
- Utilisation DeleteBucketInventoryConfiguration avec un AWS SDK ou une CLI
- Utilisation DeleteBucketLifecycle avec un AWS SDK ou une CLI
- Utilisation DeleteBucketMetricsConfiguration avec un AWS SDK ou une CLI
- Utilisation DeleteBucketPolicy avec un AWS SDK ou une CLI
- Utilisation DeleteBucketReplication avec un AWS SDK ou une CLI
- Utilisation DeleteBucketTagging avec un AWS SDK ou une CLI
- Utilisation DeleteBucketWebsite avec un AWS SDK ou une CLI
- Utilisation DeleteObject avec un AWS SDK ou une CLI
- Utilisation DeleteObjectTagging avec un AWS SDK ou une CLI
- Utilisation DeleteObjects avec un AWS SDK ou une CLI
- Utilisation DeletePublicAccessBlock avec un AWS SDK ou une CLI
- Utilisation GetBucketAccelerateConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketAcl avec un AWS SDK ou une CLI
- Utilisation GetBucketAnalyticsConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketCors avec un AWS SDK ou une CLI
- Utilisation GetBucketEncryption avec un AWS SDK ou une CLI
- Utilisation GetBucketInventoryConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketLifecycleConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketLocation avec un AWS SDK ou une CLI
- Utilisation GetBucketLogging avec un AWS SDK ou une CLI
- Utilisation GetBucketMetricsConfiguration avec un AWS SDK ou une CLI
- Utilisation GetBucketNotification avec un AWS SDK ou une CLI
- Utilisation GetBucketPolicy avec un AWS SDK ou une CLI
- Utilisation GetBucketPolicyStatus avec un AWS SDK ou une CLI
- Utilisation GetBucketReplication avec un AWS SDK ou une CLI

- Utilisation GetBucketRequestPayment avec un AWS SDK ou une CLI
- Utilisation GetBucketTagging avec un AWS SDK ou une CLI
- Utilisation GetBucketVersioning avec un AWS SDK ou une CLI
- Utilisation GetBucketWebsite avec un AWS SDK ou une CLI
- Utilisation GetObject avec un AWS SDK ou une CLI
- Utilisation GetObjectAcl avec un AWS SDK ou une CLI
- Utilisation GetObjectLegalHold avec un AWS SDK ou une CLI
- Utilisation GetObjectLockConfiguration avec un AWS SDK ou une CLI
- Utilisation GetObjectRetention avec un AWS SDK ou une CLI
- Utilisation GetObjectTagging avec un AWS SDK ou une CLI
- Utilisation GetPublicAccessBlock avec un AWS SDK ou une CLI
- Utilisation HeadBucket avec un AWS SDK ou une CLI
- Utilisation HeadObject avec un AWS SDK ou une CLI
- Utilisation ListBucketAnalyticsConfigurations avec un AWS SDK ou une CLI
- Utilisation ListBucketInventoryConfigurations avec un AWS SDK ou une CLI
- Utilisation ListBuckets avec un AWS SDK ou une CLI
- Utilisation ListMultipartUploads avec un AWS SDK ou une CLI
- Utilisation ListObjectVersions avec un AWS SDK ou une CLI
- Utilisation ListObjects avec un AWS SDK ou une CLI
- Utilisation ListObjectsV2 avec un AWS SDK ou une CLI
- Utilisation PutBucketAccelerateConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketAcl avec un AWS SDK ou une CLI
- Utilisation PutBucketCors avec un AWS SDK ou une CLI
- Utilisation PutBucketEncryption avec un AWS SDK ou une CLI
- Utilisation PutBucketLifecycleConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketLogging avec un AWS SDK ou une CLI
- Utilisation PutBucketNotification avec un AWS SDK ou une CLI
- Utilisation PutBucketNotificationConfiguration avec un AWS SDK ou une CLI
- Utilisation PutBucketPolicy avec un AWS SDK ou une CLI

- Utilisation PutBucketReplication avec un AWS SDK ou une CLI
- Utilisation PutBucketRequestPayment avec un AWS SDK ou une CLI
- Utilisation PutBucketTagging avec un AWS SDK ou une CLI
- Utilisation PutBucketVersioning avec un AWS SDK ou une CLI
- Utilisation PutBucketWebsite avec un AWS SDK ou une CLI
- Utilisation PutObject avec un AWS SDK ou une CLI
- Utilisation PutObjectAcl avec un AWS SDK ou une CLI
- Utilisation PutObjectLegalHold avec un AWS SDK ou une CLI
- Utilisation PutObjectLockConfiguration avec un AWS SDK ou une CLI
- Utilisation PutObjectRetention avec un AWS SDK ou une CLI
- Utilisation RestoreObject avec un AWS SDK ou une CLI
- Utilisation SelectObjectContent avec un AWS SDK ou une CLI
- Utilisation UploadPart avec un AWS SDK ou une CLI

Utilisation **`AbortMultipartUpload`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`AbortMultipartUpload`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Supprimer les téléchargements partitionnés incomplets

CLI

AWS CLI

Pour annuler le téléchargement partitionné spécifié

La `abort-multipart-upload` commande suivante interrompt le téléchargement partitionné de la clé `multipart/01` dans le compartiment. `my-bucket`

```
aws s3api abort-multipart-upload \
```

```
    --bucket my-bucket \
    --key multipart/01 \
    --upload-id
dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3U
```

L'ID de téléchargement requis par cette commande est généré par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`.

- Pour plus de détails sur l'API, reportez-vous AbortMultipartUploadà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande annule les téléchargements partitionnés créés il y a moins de 5 jours.

```
Remove-S3MultipartUpload -BucketName test-files -DaysBefore 5
```

Exemple 2 : Cette commande annule les téléchargements partitionnés créés avant le 2 janvier 2014.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "Thursday, January
 02, 2014"
```

Exemple 3 : Cette commande annule les téléchargements partitionnés créés avant le 2 janvier 2014, 10:45:37.

```
Remove-S3MultipartUpload -BucketName test-files -InitiatedDate "2014/01/02
 10:45:37"
```

- Pour plus de détails sur l'API, reportez-vous AbortMultipartUploadà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **AbortMultipartUploads** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser`AbortMultipartUploads`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Transfer;

/// <summary>
/// This example shows how to use the Amazon Simple Storage Service
/// (Amazon S3) to stop a multi-part upload process using the Amazon S3
/// TransferUtility.
/// </summary>
public class AbortMPU
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the S3 client object's constructor.
        // For example: RegionEndpoint.USWest2.
        IAmazonS3 client = new AmazonS3Client();

        await AbortMPUAsync(client, bucketName);
    }

    /// <summary>
    /// Cancels the multi-part copy process.
    /// </summary>
```

```
        /// <param name="client">The initialized client object used to create
        /// the TransferUtility object.</param>
        /// <param name="bucketName">The name of the S3 bucket where the
        /// multi-part copy operation is in progress.</param>
        public static async Task AbortMPUAsync(IAmazonS3 client, string
bucketName)
        {
            try
            {
                var transferUtility = new TransferUtility(client);

                // Cancel all in-progress uploads initiated before the specified
date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error: {e.Message}");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous AbortMultipartUploadsà la section Référence des AWS SDK for .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CompleteMultipartUpload`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`CompleteMultipartUpload`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Effectuer une copie en plusieurs parties

- Réalisation d'un chargement partitionné
- Utiliser les totaux de contrôle

CLI

AWS CLI

La commande suivante effectue un téléchargement en plusieurs parties pour la clé contenue `multipart/01` dans le compartiment `my-bucket` :

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-id
 dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3U
```

L'ID de téléchargement requis par cette commande est généré par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`.

L'option de téléchargement partitionné de la commande ci-dessus utilise une structure JSON qui décrit les parties du téléchargement partitionné qui doivent être réassemblées dans le fichier complet. Dans cet exemple, le `file://` préfixe est utilisé pour charger la structure JSON à partir d'un fichier du dossier local nommé`mpustruct`.

structure :

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

La valeur ETag pour chaque partie est chargée est affichée chaque fois que vous téléchargez une partie à l'aide de la `upload-part` commande et peut également être récupérée en appelant `list-parts` ou calculée en utilisant la somme de contrôle MD5 de chaque partie.

Sortie :

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "my-bucket",
    "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- Pour plus de détails sur l'API, reportez-vous CompleteMultipartUploadà la section Référence des AWS CLI commandes.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    let _complete_multipart_upload_res = client
        .complete_multipart_upload()
        .bucket(&bucket_name)
        .key(&key)
        .multipart_upload(completed_multipart_upload)
        .upload_id(upload_id)
        .send()
        .await
        .unwrap();
```

- Pour plus de détails sur l'API, voir CompleteMultipartUploadla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CopyObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`CopyObject`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Démarrer avec les compartiments et les objets
- Démarrer avec le chiffrement

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    public class CopyObject
    {
        public static async Task Main()
        {
            // Specify the AWS Region where your buckets are located if it is
            // different from the AWS Region of the default user.
            IAmazonS3 s3Client = new AmazonS3Client();

            // Remember to change these values to refer to your Amazon S3
objects.
```

```
            string sourceBucketName = "doc-example-bucket1";
            string destinationBucketName = "doc-example-bucket2";
            string sourceObjectKey = "testfile.txt";
            string destinationObjectKey = "testfilecopy.txt";

            Console.WriteLine($"Copying {sourceObjectKey} from {sourceBucketName}
 to ");
            Console.WriteLine($"{destinationBucketName} as
 {destinationObjectKey}");

            var response = await CopyingObjectAsync(
                s3Client,
                sourceObjectKey,
                destinationObjectKey,
                sourceBucketName,
                destinationBucketName);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("\nCopy complete.");
            }
        }

        /// <summary>
        /// This method calls the AWS SDK for .NET to copy an
        /// object from one Amazon S3 bucket to another.
        /// </summary>
        /// <param name="client">The Amazon S3 client object.</param>
        /// <param name="sourceKey">The name of the object to be copied.</param>
        /// <param name="destinationKey">The name under which to save the copy.</
param>
        /// <param name="sourceBucketName">The name of the Amazon S3 bucket
        /// where the file is located now.</param>
        /// <param name="destinationBucketName">The name of the Amazon S3
        /// bucket where the copy should be saved.</param>
        /// <returns>Returns a CopyObjectResponse object with the results from
        /// the async call.</returns>
        public static async Task<CopyObjectResponse> CopyingObjectAsync(
            IAmazonS3 client,
            string sourceKey,
            string destinationKey,
            string sourceBucketName,
            string destinationBucketName)
        {
```

```
            var response = new CopyObjectResponse();
            try
            {
                var request = new CopyObjectRequest
                {
                    SourceBucket = sourceBucketName,
                    SourceKey = sourceKey,
                    DestinationBucket = destinationBucketName,
                    DestinationKey = destinationKey,
                };
                response = await client.CopyObjectAsync(request);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error copying object: '{ex.Message}'");
            }

            return response;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
```

```
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::copyObject(const Aws::String &objectKey, const Aws::String
 &fromBucket, const Aws::String &toBucket,
                             const Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CopyObjectRequest request;

    request.WithCopySource(fromBucket + "/" + objectKey)
            .WithKey(objectKey)
            .WithBucket(toBucket);

    Aws::S3::Model::CopyObjectOutcome outcome = client.CopyObject(request);
    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: copyObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;

    } else {
        std::cout << "Successfully copied " << objectKey << " from " <<
 fromBucket <<
                  " to " << toBucket << "." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante copie un objet depuis `bucket-1` vers `bucket-2` :

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --bucket
 bucket-2
```

Sortie :

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
```

```
}



// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
 string, objectKey string) error {
 _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
  Bucket:     aws.String(destinationBucket),
  CopySource: aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
  Key:        aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
   sourceBucket, objectKey, destinationBucket, objectKey, err)
 }
 return err
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Copiez un objet en utilisant un S3Client.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.services.s3.model.CopyObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class CopyObject {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <objectKey> <fromBucket> <toBucket>

                Where:
                    objectKey - The name of the object (for example, book.pdf).
                    fromBucket - The S3 bucket name that contains the object (for
 example, bucket1).
                    toBucket - The S3 bucket to copy the object to (for example,
 bucket2).
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String objectKey = args[0];
        String fromBucket = args[1];
        String toBucket = args[2];
        System.out.format("Copying object %s from bucket %s to %s\n", objectKey,
 fromBucket, toBucket);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        copyBucketObject(s3, fromBucket, objectKey, toBucket);
        s3.close();
    }
```

```
    public static String copyBucketObject(S3Client s3, String fromBucket, String
 objectKey, String toBucket) {
        CopyObjectRequest copyReq = CopyObjectRequest.builder()
                .sourceBucket(fromBucket)
                .sourceKey(objectKey)
                .destinationBucket(toBucket)
                .destinationKey(objectKey)
                .build();

        try {
            CopyObjectResponse copyRes = s3.copyObject(copyReq);
            return copyRes.copyObjectResult().toString();

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

Utilisez un S3 TransferManager pour copier un objet d'un compartiment vers un autre. Consultez le fichier complet et le test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedCopy;
import software.amazon.awssdk.transfer.s3.model.Copy;
import software.amazon.awssdk.transfer.s3.model.CopyRequest;

import java.util.UUID;

    public String copyObject(S3TransferManager transferManager, String
 bucketName,
            String key, String destinationBucket, String destinationKey) {
        CopyObjectRequest copyObjectRequest = CopyObjectRequest.builder()
                .sourceBucket(bucketName)
                .sourceKey(key)
                .destinationBucket(destinationBucket)
```

```
                .destinationKey(destinationKey)
                .build();

        CopyRequest copyRequest = CopyRequest.builder()
                .copyObjectRequest(copyObjectRequest)
                .build();

        Copy copy = transferManager.copy(copyRequest);

        CompletedCopy completedCopy = copy.completionFuture().join();
        return completedCopy.response().copyObjectResult().eTag();
    }
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Copiez l'objet.

```
import { S3Client, CopyObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new CopyObjectCommand({
    CopySource: "SOURCE_BUCKET/SOURCE_OBJECT_KEY",
    Bucket: "DESTINATION_BUCKET",
    Key: "NEW_OBJECT_KEY",
  });

  try {
    const response = await client.send(command);
```

```
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun copyBucketObject(
    fromBucket: String,
    objectKey: String,
    toBucket: String,
) {
    var encodedUrl = ""
    try {
        encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
 StandardCharsets.UTF_8.toString())
    } catch (e: UnsupportedEncodingException) {
        println("URL could not be encoded: " + e.message)
    }

    val request =
        CopyObjectRequest {
            copySource = encodedUrl
            bucket = toBucket
            key = objectKey
        }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.copyObject(request)
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Copie simple d'un objet.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $folder = "copied-folder";
            $this->s3client->copyObject([
                'Bucket' => $this->bucketName,
                'CopySource' => "$this->bucketName/$fileName",
                'Key' => "$folder/$fileName-copy",
            ]);
            echo "Copied $fileName to $folder/$fileName-copy.\n";
        } catch (Exception $exception) {
            echo "Failed to copy $fileName with error: " . $exception-
>getMessage();
            exit("Please fix error with object copying before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for PHP API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande copie l'objet « sample.txt » du bucket « test-files » vers le même bucket mais avec une nouvelle clé « sample-copy.txt ».

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt
```

Exemple 2 : Cette commande copie l'objet « sample.txt » du bucket « test-files » vers le bucket « backup-files » avec la clé « sample-copy.txt ».

```
Copy-S3Object -BucketName test-files -Key sample.txt -DestinationKey sample-
copy.txt -DestinationBucket backup-files
```

Exemple 3 : Cette commande télécharge l'objet « sample.txt » du bucket « test-files » vers un fichier local nommé « local-sample.txt ».

```
Copy-S3Object -BucketName test-files -Key sample.txt -LocalFile local-sample.txt
```

Exemple 4 : télécharge l'objet unique dans le fichier spécifié. Le fichier téléchargé se trouve à l'adresse c:\downloads\data\archive.zip

```
Copy-S3Object -BucketName test-files -Key data/archive.zip -LocalFolder c:
\downloads
```

Exemple 5 : télécharge tous les objets correspondant au préfixe de clé spécifié dans le dossier local. La hiérarchie des clés relative sera préservée sous forme de sous-dossiers dans l'emplacement de téléchargement global.

```
Copy-S3Object -BucketName test-files -KeyPrefix data -LocalFolder c:\downloads
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    def copy(self, dest_object):
        """
        Copies the object to another bucket.

        :param dest_object: The destination object initialized with a bucket and
 key.
                            This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
                CopySource={"Bucket": self.object.bucket_name, "Key":
 self.object.key}
            )
            dest_object.wait_until_exists()
            logger.info(
                "Copied object from %s:%s to %s:%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
```

```
                dest_object.key,
            )
        except ClientError:
            logger.exception(
                "Couldn't copy object from %s/%s to %s/%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
            raise
```

- Pour plus de détails sur l'API, consultez CopyObjectle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Copiez un objet.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end
```

```
  # Copy the source object to the specified target bucket and rename it with the
 target key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's
 why: #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Copier un objet et ajouter le chiffrement côté serveur à l'objet de destination.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
```

```
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                        copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the
 target key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
 server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's
 why: #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "doc-example-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "doc-example-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key,
 target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key} and "\
```

```
       "encrypted the target with #{target_object.server_side_encryption}
 encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn copy_object(
    client: &Client,
    bucket_name: &str,
    object_key: &str,
    target_key: &str,
) -> Result<CopyObjectOutput, SdkError<CopyObjectError>> {
    let mut source_bucket_and_object: String = "".to_owned();
    source_bucket_and_object.push_str(bucket_name);
    source_bucket_and_object.push('/');
    source_bucket_and_object.push_str(object_key);

    client
        .copy_object()
        .copy_source(source_bucket_and_object)
        .bucket(bucket_name)
        .key(target_key)
        .send()
        .await
}
```

- Pour plus de détails sur l'API, voir CopyObjectla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
TRY.
    lo_s3->copyobject(
      iv_bucket = iv_dest_bucket
      iv_key = iv_dest_object
      iv_copysource = |{ iv_src_bucket }/{ iv_src_object }|
    ).
    MESSAGE 'Object copied to another bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d’être modifiée.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public func copyFile(from sourceBucket: String, name: String, to destBucket:
 String) async throws {
        let srcUrl = ("\(sourceBucket)/
\(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)

        let input = CopyObjectInput(
            bucket: destBucket,
            copySource: srcUrl,
            key: name
        )
        _ = try await client.copyObject(input: input)
    }
```

- Pour plus de détails sur l'API, reportez-vous CopyObjectà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateBucket`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`CreateBucket`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Démarrer avec les compartiments et les objets
- Utiliser les objets soumis au contrôle de version

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Shows how to create a new Amazon S3 bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the bucket to create.</param>
        /// <returns>A boolean value representing the success or failure of
        /// the bucket creation process.</returns>
        public static async Task<bool> CreateBucketAsync(IAmazonS3 client, string
bucketName)
        {
            try
            {
                var request = new PutBucketRequest
                {
                    BucketName = bucketName,
                    UseClientRegion = true,
                };

                var response = await client.PutBucketAsync(request);
                return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error creating bucket: '{ex.Message}'");
                return false;
            }
        }
```

Créez un compartiment avec le verrouillage des objets activé.

```
    /// <summary>
    /// Create a new Amazon S3 bucket with object lock actions.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to create.</param>
    /// <param name="enableObjectLock">True to enable object lock on the
bucket.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
enableObjectLock)
    {
        Console.WriteLine($"\tCreating bucket {bucketName} with object lock
{enableObjectLock}.");
        try
        {
            var request = new PutBucketRequest
            {
                BucketName = bucketName,
                UseClientRegion = true,
                ObjectLockEnabledForBucket = enableObjectLock,
            };

            var response = await _amazonS3.PutBucketAsync(request);

            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error creating bucket: '{ex.Message}'");
            return false;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
```

```
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally
 unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi
```

```
  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::createBucket(const Aws::String &bucketName,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::CreateBucketRequest request;
    request.SetBucket(bucketName);

    if (clientConfig.region != "us-east-1") {
        Aws::S3::Model::CreateBucketConfiguration createBucketConfig;
        createBucketConfig.SetLocationConstraint(

 Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
                        clientConfig.region));
        request.SetCreateBucketConfiguration(createBucketConfig);
    }

    Aws::S3::Model::CreateBucketOutcome outcome = client.CreateBucket(request);
    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: createBucket: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Created bucket " << bucketName <<
                  " in the specified AWS Region." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Exemple 1 : pour créer un bucket

L'`create-bucket`exemple suivant crée un compartiment nommé `my-bucket` :

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1
```

Sortie :

```
{
    "Location": "/my-bucket"
}
```

Pour plus d'informations, consultez la section Création d'un compartiment dans le guide de l'utilisateur Amazon S3.

Exemple 2 : pour créer un bucket avec l'obligation du propriétaire

L'`create-bucket`exemple suivant crée un bucket nommé `my-bucket` qui utilise le paramètre imposé par le propriétaire du bucket pour S3 Object Ownership.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

Sortie :

```
{
    "Location": "/my-bucket"
}
```

Pour plus d'informations, veuillez consulter la rubrique Contrôle de la propriété des objets et désactivation des listes ACL dans le Guide de l'utilisateur Amazon S3.

Exemple 3 : Pour créer un bucket en dehors de la région ``us-east-1``

L'`create-bucket`exemple suivant crée un compartiment nommé `my-bucket` dans la `eu-west-1` région. Les régions situées en dehors de `us-east-1` doivent être spécifiées afin de créer le compartiment dans la région souhaitée. `LocationConstraint`

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

Sortie :

```
{
    "Location": "http://my-bucket.s3.amazonaws.com/"
}
```

Pour plus d'informations, consultez la section Création d'un compartiment dans le guide de l'utilisateur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez un bucket avec la configuration par défaut.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
 _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
  Bucket: aws.String(name),
  CreateBucketConfiguration: &types.CreateBucketConfiguration{
   LocationConstraint: types.BucketLocationConstraint(region),
  },
 })
 if err != nil {
  log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
   name, region, err)
 }
 return err
}
```

Créez un compartiment avec verrouillage d'objets et attendez qu'il existe.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// CreateBucketWithLock creates a new S3 bucket with optional object locking
 enabled
// and waits for the bucket to exist before returning.
```

```
func (actor S3Actions) CreateBucketWithLock(ctx context.Context, bucket string,
 region string, enableObjectLock bool) (string, error) {
 input := &s3.CreateBucketInput{
  Bucket: aws.String(bucket),
  CreateBucketConfiguration: &types.CreateBucketConfiguration{
   LocationConstraint: types.BucketLocationConstraint(region),
  },
 }

 if enableObjectLock {
  input.ObjectLockEnabledForBucket = aws.Bool(true)
 }

 _, err := actor.S3Client.CreateBucket(ctx, input)
 if err != nil {
  var owned *types.BucketAlreadyOwnedByYou
  var exists *types.BucketAlreadyExists
  if errors.As(err, &owned) {
   log.Printf("You already own bucket %s.\n", bucket)
   err = owned
  } else if errors.As(err, &exists) {
   log.Printf("Bucket %s already exists.\n", bucket)
   err = exists
  }
 } else {
  err = s3.NewBucketExistsWaiter(actor.S3Client).Wait(
   ctx, &s3.HeadBucketInput{Bucket: aws.String(bucket)}, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for bucket %s to exist.\n", bucket)
  }
 }

 return bucket, err
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez un compartiment.

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.HeadBucketRequest;
import software.amazon.awssdk.services.s3.model.HeadBucketResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import java.net.URISyntaxException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class CreateBucket {
    public static void main(String[] args) throws URISyntaxException {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The name of the bucket to create. The bucket
 name must be unique, or an error occurs.
                """;
```

```
        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Creating a bucket named %s\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        createBucket(s3, bucketName);
        s3.close();
    }

    public static void createBucket(S3Client s3Client, String bucketName) {
        try {
            S3Waiter s3Waiter = s3Client.waiter();
            CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            s3Client.createBucket(bucketRequest);
            HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            // Wait until the bucket is created and print out the response.
            WaiterResponse<HeadBucketResponse> waiterResponse =
 s3Waiter.waitUntilBucketExists(bucketRequestWait);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println(bucketName + " is ready");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Créez un compartiment avec le verrouillage des objets activé.

```
   // Create a new Amazon S3 bucket with object lock options.
   public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
       S3Waiter s3Waiter = getClient().waiter();
       CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
           .bucket(bucketName)
           .objectLockEnabledForBucket(enableObjectLock)
           .build();

       getClient().createBucket(bucketRequest);
       HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
           .bucket(bucketName)
           .build();

       // Wait until the bucket is created and print out the response.
       s3Waiter.waitUntilBucketExists(bucketRequestWait);
       System.out.println(bucketName + " is ready");
   }
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créer le compartiment.

```
import { CreateBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});
```

```
export const main = async () => {
  const command = new CreateBucketCommand({
    // The name of the bucket. Bucket names are unique and have several other
 constraints.
    // See https://docs.aws.amazon.com/AmazonS3/latest/userguide/
bucketnamingrules.html
    Bucket: "bucket-name",
  });

  try {
    const { Location } = await client.send(command);
    console.log(`Bucket created with location ${Location}`);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun createNewBucket(bucketName: String) {
    val request =
        CreateBucketRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
```

```
        println("$bucketName is ready")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez un compartiment.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $this->s3client->createBucket([
                'Bucket' => $this->bucketName,
                'CreateBucketConfiguration' => ['LocationConstraint' => $region],
            ]);
            echo "Created bucket named: $this->bucketName \n";
        } catch (Exception $exception) {
            echo "Failed to create bucket $this->bucketName with error: " .
$exception->getMessage();
            exit("Please fix error with bucket creation before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for PHP API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créer un compartiment avec les paramètres par défaut.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def create(self, region_override=None):
        """
        Create an Amazon S3 bucket in the default Region for the account or in
 the
        specified Region.

        :param region_override: The Region in which to create the bucket. If this
 is
                                not specified, the Region configured in your
 shared
                                credentials is used.
        """
        if region_override is not None:
            region = region_override
        else:
            region = self.bucket.meta.client.meta.region_name
        try:
```

```
            self.bucket.create(CreateBucketConfiguration={"LocationConstraint":
 region})

            self.bucket.wait_until_exists()
            logger.info("Created bucket '%s' in region=%s", self.bucket.name,
 region)
        except ClientError as error:
            logger.exception(
                "Couldn't create bucket named '%s' in region=%s.",
                self.bucket.name,
                region,
            )
            raise error
```

Créer un compartiment soumis à la gestion des versions avec une configuration de cycle de vie.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
 lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
```

```
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )
```

```
    return bucket
```

- Pour plus de détails sur l'API, consultez CreateBucketle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
 This is a client-side object until
  #                                 create is called.
  def initialize(bucket)
    @bucket = bucket
  end

  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
```

```
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      "None. You must create a bucket before you can get its location!"
    else
      @bucket.client.get_bucket_location(bucket:
 @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("doc-example-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn create_bucket(
    client: &Client,
    bucket_name: &str,
    region: &str,
) -> Result<CreateBucketOutput, SdkError<CreateBucketError>> {
    let constraint = BucketLocationConstraint::from(region);
    let cfg = CreateBucketConfiguration::builder()
        .location_constraint(constraint)
        .build();
    client
        .create_bucket()
        .create_bucket_configuration(cfg)
        .bucket(bucket_name)
        .send()
        .await
}
```

- Pour plus de détails sur l'API, voir CreateBucketla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        lo_s3->createbucket(
            iv_bucket = iv_bucket_name
        ).
        MESSAGE 'S3 bucket created.' TYPE 'I'.
      CATCH /aws1/cx_s3_bucketalrdyexists.
        MESSAGE 'Bucket name already exists.' TYPE 'E'.
      CATCH /aws1/cx_s3_bktalrdyownedbyyou.
        MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public func createBucket(name: String) async throws {
        let config = S3ClientTypes.CreateBucketConfiguration(
            locationConstraint: .usEast2
        )
        let input = CreateBucketInput(
            bucket: name,
            createBucketConfiguration: config
        )
        _ = try await client.createBucket(input: input)
```

```
    }
```

- Pour plus de détails sur l'API, reportez-vous CreateBucketà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateMultiRegionAccessPoint`** avec un AWS SDK ou une CLI

L'exemple de code suivant montre comment utiliser`CreateMultiRegionAccessPoint`.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Configurez le client de contrôle S3 pour envoyer une demande à la région us-west-2.

```
        suspend fun createS3ControlClient(): S3ControlClient {
            // Configure your S3ControlClient to send requests to US West
(Oregon).
            val s3Control = S3ControlClient.fromEnvironment {
                region = "us-west-2"
            }
            return s3Control
        }
```

Créez le point d'accès multirégional.

```
    suspend fun createMrap(
```

```
        s3Control: S3ControlClient,
        accountIdParam: String,
        bucketName1: String,
        bucketName2: String,
        mrapName: String,
    ): String {
        println("Creating MRAP ...")
        val createMrapResponse: CreateMultiRegionAccessPointResponse =
            s3Control.createMultiRegionAccessPoint {
                accountId = accountIdParam
                clientToken = UUID.randomUUID().toString()
                details {
                    name = mrapName
                    regions = listOf(
                        Region {
                            bucket = bucketName1
                        },
                        Region {
                            bucket = bucketName2
                        },
                    )
                }
            }
        val requestToken: String? = createMrapResponse.requestTokenArn

        // Use the request token to check for the status of the
CreateMultiRegionAccessPoint operation.
        if (requestToken != null) {
            waitForSucceededStatus(s3Control, requestToken, accountIdParam)
            println("MRAP created")
        }

        val getMrapResponse =
            s3Control.getMultiRegionAccessPoint(
                input = GetMultiRegionAccessPointRequest {
                    accountId = accountIdParam
                    name = mrapName
                },
            )
        val mrapAlias = getMrapResponse.accessPoint?.alias
        return "arn:aws:s3::$accountIdParam:accesspoint/$mrapAlias"
    }
```

Attendez que le point d'accès multirégional soit disponible.

```
        suspend fun waitForSucceededStatus(
            s3Control: S3ControlClient,
            requestToken: String,
            accountIdParam: String,
            timeBetweenChecks: Duration = 1.minutes,
        ) {
            var describeResponse: DescribeMultiRegionAccessPointOperationResponse
            describeResponse = s3Control.describeMultiRegionAccessPointOperation(
                input = DescribeMultiRegionAccessPointOperationRequest {
                    accountId = accountIdParam
                    requestTokenArn = requestToken
                },
            )

            var status: String? = describeResponse.asyncOperation?.requestStatus
            while (status != "SUCCEEDED") {
                delay(timeBetweenChecks)
                describeResponse =
s3Control.describeMultiRegionAccessPointOperation(
                    input = DescribeMultiRegionAccessPointOperationRequest {
                        accountId = accountIdParam
                        requestTokenArn = requestToken
                    },
                )
                status = describeResponse.asyncOperation?.requestStatus
                println(status)
            }
        }
```

- Pour en savoir plus, consultez Guide du développeur d'AWS SDK pour Kotlin.
- Pour plus de détails sur l'API, reportez-vous CreateMultiRegionAccessPointà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`CreateMultipartUpload`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`CreateMultipartUpload`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Effectuer une copie en plusieurs parties
- Réalisation d'un chargement partitionné
- Utiliser les totaux de contrôle

CLI

AWS CLI

La commande suivante crée un téléchargement partitionné dans le bucket `my-bucket` avec la clé `multipart/01` :

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

Sortie :

```
{
    "Bucket": "my-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
    "Key": "multipart/01"
}
```

Le fichier terminé sera nommé `01` dans un dossier appelé `multipart` dans le compartiment`my-bucket`. Enregistrez l'ID de téléchargement, la clé et le nom du compartiment à utiliser avec la `upload-part` commande.

- Pour plus de détails sur l'API, reportez-vous CreateMultipartUploadà la section Référence des AWS CLI commandes.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
let multipart_upload_res: CreateMultipartUploadOutput = client
    .create_multipart_upload()
    .bucket(&bucket_name)
    .key(&key)
    .send()
    .await
    .unwrap();
```

- Pour plus de détails sur l'API, voir CreateMultipartUploadla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucket`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucket`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Démarrer avec les compartiments et les objets

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Shows how to delete an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket to
delete.</param>
        /// <returns>A boolean value that represents the success or failure of
        /// the delete operation.</returns>
        public static async Task<bool> DeleteBucketAsync(IAmazonS3 client, string
bucketName)
        {
            var request = new DeleteBucketRequest
            {
                BucketName = bucketName,
            };

            var response = await client.DeleteBucketAsync(request);
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
```

```
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::deleteBucket(const Aws::String &bucketName,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
 {

    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
            client.DeleteBucket(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: deleteBucket: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "The bucket was deleted" << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante supprime un bucket nommé `my-bucket` :

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// DeleteBucket deletes a bucket. The bucket must be empty or an error is
 returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
```

```
 _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
  Bucket: aws.String(bucketName)})
 if err != nil {
  log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
 }
 return err
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucket)
                .build();

        s3.deleteBucket(deleteBucketRequest);
        s3.close();
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez le compartiment.

```
import { DeleteBucketCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Delete a bucket.
export const main = async () => {
  const command = new DeleteBucketCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for JavaScript API.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un compartiment vide.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $this->s3client->deleteBucket([
                'Bucket' => $this->bucketName,
            ]);
            echo "Deleted bucket $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to delete $this->bucketName with error: " . $exception-
>getMessage();
            exit("Please fix error with bucket deletion before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for PHP API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande supprime tous les objets et toutes les versions d'objets du bucket « test-files », puis supprime le bucket. La commande vous demandera une confirmation avant de continuer. Ajoutez le commutateur -Force pour supprimer la confirmation. Notez que les compartiments qui ne sont pas vides ne peuvent pas être supprimés.

```
Remove-S3Bucket -BucketName test-files -DeleteBucketContent
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def delete(self):
        """
        Delete the bucket. The bucket must be empty or an error is raised.
        """
        try:
            self.bucket.delete()
            self.bucket.wait_until_not_exists()
            logger.info("Bucket %s successfully deleted.", self.bucket.name)
        except ClientError:
            logger.exception("Couldn't delete bucket %s.", self.bucket.name)
            raise
```

- Pour plus de détails sur l'API, consultez DeleteBucketle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
 # Deletes the objects in an Amazon S3 bucket and deletes the bucket.
 #
 # @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
 def delete_bucket(bucket)
   puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?
")
   answer = gets.chomp.downcase
   if answer == "y"
     bucket.objects.batch_delete!
     bucket.delete
     puts("Emptied and deleted bucket #{bucket.name}.\n")
   end
 rescue Aws::Errors::ServiceError => e
   puts("Couldn't empty and delete bucket #{bucket.name}.")
   puts("\t#{e.code}: #{e.message}")
   raise
 end
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn delete_bucket(client: &Client, bucket_name: &str) -> Result<(),
 Error> {
    client.delete_bucket().bucket(bucket_name).send().await?;
    println!("Bucket deleted");
    Ok(())
}
```

- Pour plus de détails sur l'API, voir DeleteBucketla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.

        lo_s3->deletebucket(
            iv_bucket = iv_bucket_name
        ).
        MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
public func deleteBucket(name: String) async throws {
    let input = DeleteBucketInput(
        bucket: name
    )
    _ = try await client.deleteBucket(input: input)
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteBucketAnalyticsConfiguration** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketAnalyticsConfiguration`.

CLI

AWS CLI

Pour supprimer une configuration d'analyse pour un bucket

L'`delete-bucket-analytics-configuration`exemple suivant supprime la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande supprime le filtre d'analyse nommé « testfilter » dans le compartiment S3 donné.

```
Remove-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketAnalyticsConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketCors`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Deletes a CORS configuration from an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used
        /// to delete the CORS configuration from the bucket.</param>
        private static async Task DeleteCORSConfigurationAsync(AmazonS3Client
client)
        {
            DeleteCORSConfigurationRequest request = new
DeleteCORSConfigurationRequest()
            {
                BucketName = BucketName,
            };
            await client.DeleteCORSConfigurationAsync(request);
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketCorsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante supprime une configuration de partage de ressources entre origines d'un compartiment nommé : `my-bucket`

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketCorsà la section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def delete_cors(self):
        """
        Delete the CORS rules from the bucket.

        :param bucket_name: The name of the bucket to update.
        """
```

```
        try:
            self.bucket.Cors().delete()
            logger.info("Deleted CORS from bucket '%s'.", self.bucket.name)
        except ClientError:
            logger.exception("Couldn't delete CORS from bucket '%s'.",
 self.bucket.name)
            raise
```

- Pour plus de détails sur l'API, consultez DeleteBucketCorsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
 an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Deletes the CORS configuration of a bucket.
  #
  # @return [Boolean] True if the CORS rules were deleted; otherwise, false.
  def delete_cors
    @bucket_cors.delete
    true
```

```
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete CORS rules for #{@bucket_cors.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketCorsà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketEncryption`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketEncryption`.

CLI

AWS CLI

Pour supprimer la configuration de chiffrement côté serveur d'un bucket

L'`delete-bucket-encryption`exemple suivant supprime la configuration de chiffrement côté serveur du compartiment spécifié.

```
aws s3api delete-bucket-encryption \
    --bucket my-bucket
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketEncryptionà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cela désactive le chiffrement activé pour le compartiment S3 fourni.

```
Remove-S3BucketEncryption -BucketName 's3casetestbucket'
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketEncryption (DeleteBucketEncryption)" on
 target "s3casetestbucket".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketEncryptionà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketInventoryConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketInventoryConfiguration`.

CLI

AWS CLI

Pour supprimer la configuration d'inventaire d'un bucket

L'`delete-bucket-inventory-configuration`exemple suivant supprime la configuration d'inventaire avec l'ID 1 du compartiment spécifié.

```
aws s3api delete-bucket-inventory-configuration \
```

```
    --bucket my-bucket \
    --id 1
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketInventoryConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande supprime l'inventaire nommé « testInventoryName » correspondant au compartiment S3 donné.

```
Remove-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testInventoryName'
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketInventoryConfiguration
 (DeleteBucketInventoryConfiguration)" on target "s3testbucket".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketInventoryConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketLifecycle`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketLifecycle`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// This method removes the Lifecycle configuration from the named
        /// S3 bucket.
        /// </summary>
        /// <param name="client">The S3 client object used to call
        /// the RemoveLifecycleConfigAsync method.</param>
        /// <param name="bucketName">A string representing the name of the
        /// S3 bucket from which the configuration will be removed.</param>
        public static async Task RemoveLifecycleConfigAsync(IAmazonS3 client,
string bucketName)
        {
            var request = new DeleteLifecycleConfigurationRequest()
            {
                BucketName = bucketName,
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketLifecycleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante supprime une configuration de cycle de vie d'un compartiment nommé `my-bucket` :

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketLifecycleà la section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def delete_lifecycle_configuration(self):
        """
        Remove the lifecycle configuration from the specified bucket.
        """
        try:
            self.bucket.LifecycleConfiguration().delete()
            logger.info(
                "Deleted lifecycle configuration for bucket '%s'.",
 self.bucket.name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete lifecycle configuration for bucket '%s'.",
```

```
                self.bucket.name,
            )
            raise
```

- Pour plus de détails sur l'API, consultez DeleteBucketLifecyclele AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketMetricsConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketMetricsConfiguration`.

CLI

AWS CLI

Pour supprimer la configuration des métriques d'un bucket

L'`delete-bucket-metrics-configuration`exemple suivant supprime la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande supprime le filtre de métriques nommé « testmetrics » dans le compartiment S3 donné.

```
Remove-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testmetrics'
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketMetricsConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketPolicy`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketPolicy`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::deleteBucketPolicy(const Aws::String &bucketName,
                                    const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::DeleteBucketPolicyRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketPolicyOutcome outcome =
 client.DeleteBucketPolicy(request);
```

```
    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: deleteBucketPolicy: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Policy was deleted from the bucket." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante supprime une politique de bucket d'un bucket nommé `my-bucket` :

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.DeleteBucketPolicyRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class DeleteBucketPolicy {
    public static void main(String[] args) {

        final String usage = """

                Usage:
                    <bucketName>

                Where:
                    bucketName - The Amazon S3 bucket to delete the policy from
 (for example, bucket1).""";

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Deleting policy from bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        deleteS3BucketPolicy(s3, bucketName);
        s3.close();
    }

    // Delete the bucket policy.
    public static void deleteS3BucketPolicy(S3Client s3, String bucketName) {
        DeleteBucketPolicyRequest delReq = DeleteBucketPolicyRequest.builder()
```

```
                .bucket(bucketName)
                .build();

        try {
            s3.deleteBucketPolicy(delReq);
            System.out.println("Done!");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez la politique du compartiment.

```
import { DeleteBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// This will remove the policy from the bucket.
export const main = async () => {
  const command = new DeleteBucketPolicyCommand({
    Bucket: "test-bucket",
  });

  try {
```

```
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deleteS3BucketPolicy(bucketName: String?) {
    val request =
        DeleteBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucketPolicy(request)
        println("Done!")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande supprime la politique de compartiment associée au compartiment S3 donné.

```
Remove-S3BucketPolicy -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def delete_policy(self):
        """
        Delete the security policy from the bucket.
        """
        try:
            self.bucket.Policy().delete()
```

```
            logger.info("Deleted policy for bucket '%s'.", self.bucket.name)
        except ClientError:
            logger.exception(
                "Couldn't delete policy for bucket '%s'.", self.bucket.name
            )
            raise
```

- Pour plus de détails sur l'API, consultez DeleteBucketPolicyle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
 configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  def delete_policy
    @bucket_policy.delete
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't delete the policy from #{@bucket_policy.bucket.name}. Here's
 why: #{e.message}"
    false
  end
```

```
end
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketPolicyà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketReplication`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketReplication`.

CLI

AWS CLI

La commande suivante supprime une configuration de réplication d'un compartiment nommé `my-bucket` :

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketReplicationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Supprime la configuration de réplication associée au bucket nommé « mybucket ». Notez que cette opération nécessite une autorisation pour l'DeleteReplicationConfiguration action s3 :. Vous serez invité à confirmer avant que l'opération ne se poursuive. Pour supprimer la confirmation, utilisez le commutateur -Force.

```
Remove-S3BucketReplication -BucketName mybucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketReplicationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketTagging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketTagging`.

CLI

AWS CLI

La commande suivante supprime une configuration de balisage d'un compartiment nommé : `my-bucket`

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketTaggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande supprime toutes les balises associées au compartiment S3 donné.

```
Remove-S3BucketTagging -BucketName 's3testbucket'
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketTagging (DeleteBucketTagging)" on target
 "s3testbucket".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketTaggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteBucketWebsite`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteBucketWebsite`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::deleteBucketWebsite(const Aws::String &bucketName,
                                     const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteBucketWebsiteRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketWebsiteOutcome outcome =
            client.DeleteBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: deleteBucketWebsite: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Website configuration was removed." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante supprime la configuration d'un site Web d'un compartiment nommé `my-bucket` :

```
aws s3api delete-bucket-website --bucket my-bucket
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.DeleteBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class DeleteWebsiteConfiguration {
    public static void main(String[] args) {
        final String usage = """

                Usage:     <bucketName>

                Where:
                    bucketName - The Amazon S3 bucket to delete the website
 configuration from.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Deleting website configuration for Amazon S3 bucket:
 %s\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        deleteBucketWebsiteConfig(s3, bucketName);
        System.out.println("Done!");
        s3.close();
    }

    public static void deleteBucketWebsiteConfig(S3Client s3, String bucketName)
 {
        DeleteBucketWebsiteRequest delReq = DeleteBucketWebsiteRequest.builder()
                .bucket(bucketName)
                .build();

        try {
            s3.deleteBucketWebsite(delReq);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.out.println("Failed to delete website configuration!");
            System.exit(1);
        }
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez la configuration du site Web du compartiment.

```
import { DeleteBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Disable static website hosting on the bucket.
export const main = async () => {
  const command = new DeleteBucketWebsiteCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande désactive la propriété d'hébergement statique du site Web du compartiment S3 donné.

```
Remove-S3BucketWebsite -BucketName 's3testbucket'
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3BucketWebsite (DeleteBucketWebsite)" on target
 "s3testbucket".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
```

- Pour plus de détails sur l'API, reportez-vous DeleteBucketWebsiteà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteObject`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Utiliser les objets soumis au contrôle de version

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un objet dans un compartiment S3 non soumis à la gestion des versions.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example shows how to delete an object from a non-versioned Amazon
    /// Simple Storage Service (Amazon S3) bucket.
    /// </summary>
    public class DeleteObject
    {
        /// <summary>
        /// The Main method initializes the necessary variables and then calls
        /// the DeleteObjectNonVersionedBucketAsync method to delete the object
        /// named by the keyName parameter.
        /// </summary>
        public static async Task Main()
        {
            const string bucketName = "doc-example-bucket";
            const string keyName = "testfile.txt";

            // If the Amazon S3 bucket is located in an AWS Region other than the
            // Region of the default account, define the AWS Region for the
            // Amazon S3 bucket in your call to the AmazonS3Client constructor.
            // For example RegionEndpoint.USWest2.
            IAmazonS3 client = new AmazonS3Client();
            await DeleteObjectNonVersionedBucketAsync(client, bucketName,
keyName);
        }

        /// <summary>
```

```
        /// The DeleteObjectNonVersionedBucketAsync takes care of deleting the
        /// desired object from the named bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client used to delete
        /// an object from an Amazon S3 bucket.</param>
        /// <param name="bucketName">The name of the bucket from which the
        /// object will be deleted.</param>
        /// <param name="keyName">The name of the object to delete.</param>
        public static async Task DeleteObjectNonVersionedBucketAsync(IAmazonS3
client, string bucketName, string keyName)
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                };

                Console.WriteLine($"Deleting object: {keyName}");
                await client.DeleteObjectAsync(deleteObjectRequest);
                Console.WriteLine($"Object: {keyName} deleted from
{bucketName}.");
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error encountered on server.
Message:'{ex.Message}' when deleting an object.");
            }
        }
    }
```

Supprimez un objet dans un compartiment S3 soumis à la gestion des versions.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example creates an object in an Amazon Simple Storage Service
```

```
    /// (Amazon S3) bucket and then deletes the object version that was
    /// created.
    /// </summary>
    public class DeleteObjectVersion
    {
        public static async Task Main()
        {
            string bucketName = "doc-example-bucket";
            string keyName = "verstioned-object.txt";

            // If the AWS Region of the default user is different from the AWS
            // Region of the Amazon S3 bucket, pass the AWS Region of the
            // bucket region to the Amazon S3 client object's constructor.
            // Define it like this:
            //      RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
            IAmazonS3 client = new AmazonS3Client();

            await CreateAndDeleteObjectVersionAsync(client, bucketName, keyName);
        }

        /// <summary>
        /// This method creates and then deletes a versioned object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
        /// create and delete the object.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket where the
        /// object will be created and deleted.</param>
        /// <param name="keyName">The key name of the object to create.</param>
        public static async Task CreateAndDeleteObjectVersionAsync(IAmazonS3
client, string bucketName, string keyName)
        {
            try
            {
                // Add a sample object.
                string versionID = await PutAnObject(client, bucketName,
keyName);

                // Delete the object by specifying an object key and a version
ID.
                DeleteObjectRequest request = new DeleteObjectRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                    VersionId = versionID,
```

```
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(request);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }

        /// <summary>
        /// This method is used to create the temporary Amazon S3 object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 object which will be
 used
        /// to create the temporary Amazon S3 object.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket where the
 object
        /// will be created.</param>
        /// <param name="objectKey">The name of the Amazon S3 object co create.</
param>
        /// <returns>The Version ID of the created object.</returns>
        public static async Task<string> PutAnObject(IAmazonS3 client, string
 bucketName, string objectKey)
        {
            PutObjectRequest request = new PutObjectRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
                ContentBody = "This is the content body!",
            };

            PutObjectResponse response = await client.PutObjectAsync(request);
            return response.VersionId;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function delete_item_in_bucket
#
# This function deletes the specified file from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - The key (file name) in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
#############################################################################
function delete_item_in_bucket() {
  local bucket_name=$1
  local key=$2
  local response

  response=$(aws s3api delete-object \
    --bucket "$bucket_name" \
    --key "$key")

  # shellcheck disable=SC2181
```

```
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n
$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::deleteObject(const Aws::String &objectKey,
                              const Aws::String &fromBucket,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteObjectRequest request;

    request.WithKey(objectKey)
            .WithBucket(fromBucket);

    Aws::S3::Model::DeleteObjectOutcome outcome =
            client.DeleteObject(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error: deleteObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Successfully deleted the object." << std::endl;
```

```
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante supprime un objet nommé `test.txt` dans un compartiment nommé `my-bucket` :

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

Si le versionnement des compartiments est activé, la sortie contiendra l'ID de version du marqueur de suppression :

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
  "DeleteMarker": true
}
```

Pour plus d'informations sur la suppression d'objets, consultez Supprimer des objets dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// DeleteObject deletes an object from a bucket.
func (actor S3Actions) DeleteObject(ctx context.Context, bucket string, key
 string, versionId string, bypassGovernance bool) (bool, error) {
 deleted := false
 input := &s3.DeleteObjectInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
 }
 if versionId != "" {
  input.VersionId = aws.String(versionId)
 }
 if bypassGovernance {
  input.BypassGovernanceRetention = aws.Bool(true)
 }
 _, err := actor.S3Client.DeleteObject(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in %s.\n", key, bucket)
   err = noKey
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
```

```
   case "AccessDenied":
    log.Printf("Access denied: cannot delete object %s from %s.\n", key, bucket)
    err = nil
   case "InvalidArgument":
    if bypassGovernance {
     log.Printf("You cannot specify bypass governance on a bucket without lock
 enabled.")
     err = nil
    }
   }
  }
 } else {
  deleted = true
 }
 return deleted, err
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS SDK for Go API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un objet.

```
import { DeleteObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new DeleteObjectCommand({
    Bucket: "test-bucket",
    Key: "test-key.txt",
```

```
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un objet.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    def delete(self):
        """
        Deletes the object.
```

```
        """
        try:
            self.object.delete()
            self.object.wait_until_not_exists()
            logger.info(
                "Deleted object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't delete object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
            raise
```

Faire revenir un objet à une version antérieure en supprimant les versions ultérieures de l'objet.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
```

```
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
            else:
                break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
 f"{object_key}."
        )
```

Faire revenir un objet supprimé en supprimant le marqueur de suppression actif de l'objet.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
 marker.
    By removing the delete marker, we make the previous version the latest
 version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.
```

```
    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
                latest_version["LastModified"],
            )
            obj = bucket.Object(object_key)
            obj.Version(latest_version["VersionId"]).delete()
            logger.info(
                "Revived %s, active version is now %s  with body '%s'",
                object_key,
                obj.version_id,
                obj.get()["Body"].read(),
            )
        else:
            logger.warning(
                "Delete marker is not the latest version for %s!", object_key
            )
    elif "Versions" in response:
        logger.warning("Got an active version for %s, nothing to do.",
 object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Créez un gestionnaire Lambda qui supprime un marqueur de suppression d'un objet S3. Ce gestionnaire peut être utilisé pour nettoyer efficacement les marqueurs de suppression superflus dans un compartiment soumis au contrôle de version.

```
import logging
from urllib import parse
```

```
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")


def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of
 the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]

    try:
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
 obj_key
        )

        try:
```

```
            # If this call does not raise an error, the object version is not a
 delete
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "PermanentFailure"
            result_string = (
                f"Object {obj_key}, ID {obj_version_id} is not " f"a delete
 marker."
            )

            logger.debug(response)
            logger.warning(result_string)
        except ClientError as error:
            delete_marker = error.response["ResponseMetadata"]
["HTTPHeaders"].get(
                "x-amz-delete-marker", "false"
            )
            if delete_marker == "true":
                logger.info(
                    "Object %s, version %s is a delete marker.", obj_key,
 obj_version_id
                )
                try:
                    s3.delete_object(
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                    )
                    result_code = "Succeeded"
                    result_string = (
                        f"Successfully removed delete marker "
                        f"{obj_version_id} from object {obj_key}."
                    )
                    logger.info(result_string)
                except ClientError as error:
                    # Mark request timeout as a temporary failure so it will be
 retried.
                    if error.response["Error"]["Code"] == "RequestTimeout":
                        result_code = "TemporaryFailure"
                        result_string = (
                            f"Attempt to remove delete marker from  "
                            f"object {obj_key} timed out."
                        )
                        logger.info(result_string)
```

```
                    else:
                        raise
            else:
                raise ValueError(
                    f"The x-amz-delete-marker header is either not "
                    f"present or is not 'true'."
                )
    except Exception as error:
        # Mark all other exceptions as permanent failures.
        result_code = "PermanentFailure"
        result_string = str(error)
        logger.exception(error)
    finally:
        results.append(
            {
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

- Pour plus de détails sur l'API, consultez DeleteObjectle AWS manuel de référence de l'API SDK for Python (Boto3).

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn remove_object(client: &Client, bucket: &str, key: &str) -> Result<(),
 Error> {
    client
        .delete_object()
        .bucket(bucket)
        .key(key)
        .send()
        .await?;

    println!("Object deleted.");

    Ok(())
}
```

- Pour plus de détails sur l'API, voir DeleteObjectla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        lo_s3->deleteobject(
            iv_bucket = iv_bucket_name
            iv_key = iv_object_key
        ).
        MESSAGE 'Object deleted from S3 bucket.' TYPE 'I'.
      CATCH /aws1/cx_s3_nosuchbucket.
        MESSAGE 'Bucket does not exist.' TYPE 'E'.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
public func deleteFile(bucket: String, key: String) async throws {
    let input = DeleteObjectInput(
        bucket: bucket,
        key: key
    )

    do {
        _ = try await client.deleteObject(input: input)
    } catch {
        throw error
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteObjectTagging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteObjectTagging`.

CLI

AWS CLI

Pour supprimer les ensembles de balises d'un objet

L'`delete-object-tagging`exemple suivant supprime de l'objet `doc1.rtf` la balise avec la clé spécifiée.

```
aws s3api delete-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeleteObjectTaggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande supprime toutes les balises associées à l'objet avec la clé « testfile.txt » dans le compartiment S3 donné.

```
Remove-S3ObjectTagSet -Key 'testfile.txt' -BucketName 's3testbucket' -Select
 '^Key'
```

Sortie :

```
Confirm
Are you sure you want to perform this action?
Performing the operation "Remove-S3ObjectTagSet (DeleteObjectTagging)" on target
 "testfile.txt".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is
 "Y"): Y
testfile.txt
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectTaggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteObjects`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeleteObjects`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Démarrer avec les compartiments et les objets

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez tous les objets d'un compartiment S3.

```
        /// <summary>
        /// Delete all of the objects stored in an existing Amazon S3 bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the bucket from which the
        /// contents will be deleted.</param>
        /// <returns>A boolean value that represents the success or failure of
        /// deleting all of the objects in the bucket.</returns>
        public static async Task<bool> DeleteBucketContentsAsync(IAmazonS3
client, string bucketName)
```

```
        {
            // Iterate over the contents of the bucket and delete all objects.
            var request = new ListObjectsV2Request
            {
                BucketName = bucketName,
            };

            try
            {
                ListObjectsV2Response response;

                do
                {
                    response = await client.ListObjectsV2Async(request);
                    response.S3Objects
                        .ForEach(async obj => await
client.DeleteObjectAsync(bucketName, obj.Key));

                    // If the response is truncated, set the request
ContinuationToken
                    // from the NextContinuationToken property of the response.
                    request.ContinuationToken = response.NextContinuationToken;
                }
                while (response.IsTruncated);

                return true;
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error deleting objects: {ex.Message}");
                return false;
            }
        }
```

Supprimez plusieurs objets dans un compartiment S3 non soumis à la gestion des versions.

```
    using System;
    using System.Collections.Generic;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;
```

```
   /// <summary>
   /// This example shows how to delete multiple objects from an Amazon Simple
   /// Storage Service (Amazon S3) bucket.
   /// </summary>
   public class DeleteMultipleObjects
   {
       /// <summary>
       /// The Main method initializes the Amazon S3 client and the name of
       /// the bucket and then passes those values to MultiObjectDeleteAsync.
       /// </summary>
       public static async Task Main()
       {
           const string bucketName = "doc-example-bucket";

           // If the Amazon S3 bucket from which you wish to delete objects is
not
           // located in the same AWS Region as the default user, define the
           // AWS Region for the Amazon S3 bucket as a parameter to the client
           // constructor.
           IAmazonS3 s3Client = new AmazonS3Client();

           await MultiObjectDeleteAsync(s3Client, bucketName);
       }

       /// <summary>
       /// This method uses the passed Amazon S3 client to first create and then
       /// delete three files from the named bucket.
       /// </summary>
       /// <param name="client">The initialized Amazon S3 client object used to
call
       /// Amazon S3 methods.</param>
       /// <param name="bucketName">The name of the Amazon S3 bucket where
objects
       /// will be created and then deleted.</param>
       public static async Task MultiObjectDeleteAsync(IAmazonS3 client, string
bucketName)
       {
           // Create three sample objects which we will then delete.
           var keysAndVersions = await PutObjectsAsync(client, 3, bucketName);

           // Now perform the multi-object delete, passing the key names and
           // version IDs. Since we are working with a non-versioned bucket,
           // the object keys collection includes null version IDs.
```

```
            DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest
            {
                BucketName = bucketName,
                Objects = keysAndVersions,
            };

            // You can add a specific object key to the delete request using the
            // AddKey method of the multiObjectDeleteRequest.
            try
            {
                DeleteObjectsResponse response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
                Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
            }
            catch (DeleteObjectsException e)
            {
                PrintDeletionErrorStatus(e);
            }
        }

        /// <summary>
        /// Prints the list of errors raised by the call to DeleteObjectsAsync.
        /// </summary>
        /// <param name="ex">A collection of exceptions returned by the call to
        /// DeleteObjectsAsync.</param>
        public static void PrintDeletionErrorStatus(DeleteObjectsException ex)
        {
            DeleteObjectsResponse errorResponse = ex.Response;
            Console.WriteLine("x {0}", errorResponse.DeletedObjects.Count);

            Console.WriteLine($"Successfully deleted
{errorResponse.DeletedObjects.Count}.");
            Console.WriteLine($"No. of objects failed to delete =
{errorResponse.DeleteErrors.Count}");

            Console.WriteLine("Printing error data...");
            foreach (DeleteError deleteError in errorResponse.DeleteErrors)
            {
                Console.WriteLine($"Object Key:
{deleteError.Key}\t{deleteError.Code}\t{deleteError.Message}");
            }
        }
```

```
        /// <summary>
        /// This method creates simple text file objects that can be used in
        /// the delete method.
        /// </summary>
        /// <param name="client">The Amazon S3 client used to call
PutObjectAsync.</param>
        /// <param name="number">The number of objects to create.</param>
        /// <param name="bucketName">The name of the bucket where the objects
        /// will be created.</param>
        /// <returns>A list of keys (object keys) and versions that the calling
        /// method will use to delete the newly created files.</returns>
        public static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
client, int number, string bucketName)
        {
            List<KeyVersion> keys = new List<KeyVersion>();
            for (int i = 0; i < number; i++)
            {
                string key = "ExampleObject-" + new System.Random().Next();
                PutObjectRequest request = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = key,
                    ContentBody = "This is the content body!",
                };

                PutObjectResponse response = await
client.PutObjectAsync(request);

                // For non-versioned bucket operations, we only need the
                // object key.
                KeyVersion keyVersion = new KeyVersion
                {
                    Key = key,
                };
                keys.Add(keyVersion);
            }

            return keys;
        }
    }
```

Supprimez plusieurs objets dans un compartiment S3 soumis à la gestion des versions.

```
   using System;
   using System.Collections.Generic;
   using System.Threading.Tasks;
   using Amazon.S3;
   using Amazon.S3.Model;

   /// <summary>
   /// This example shows how to delete objects in a version-enabled Amazon
   /// Simple StorageService (Amazon S3) bucket.
   /// </summary>
   public class DeleteMultipleObjects
   {
       public static async Task Main()
       {
           string bucketName = "doc-example-bucket";

           // If the AWS Region for your Amazon S3 bucket is different from
           // the AWS Region of the default user, define the AWS Region for
           // the Amazon S3 bucket and pass it to the client constructor
           // like this:
           // RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
           IAmazonS3 s3Client;

           s3Client = new AmazonS3Client();
           await DeleteMultipleObjectsFromVersionedBucketAsync(s3Client,
bucketName);
       }

       /// <summary>
       /// This method removes multiple versions and objects from a
       /// version-enabled Amazon S3 bucket.
       /// </summary>
       /// <param name="client">The initialized Amazon S3 client object used to
call
       /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
       /// RemoveDeleteMarkersAsync.</param>
       /// <param name="bucketName">The name of the bucket from which to delete
       /// objects.</param>
       public static async Task
DeleteMultipleObjectsFromVersionedBucketAsync(IAmazonS3 client, string
bucketName)
       {
```

```
            // Delete objects (specifying object version in the request).
            await DeleteObjectVersionsAsync(client, bucketName);

            // Delete objects (without specifying object version in the request).
            var deletedObjects = await DeleteObjectsAsync(client, bucketName);

            // Additional exercise - remove the delete markers Amazon S3 returned
from
            // the preceding response. This results in the objects reappearing
            // in the bucket (you can verify the appearance/disappearance of
            // objects in the console).
            await RemoveDeleteMarkersAsync(client, bucketName, deletedObjects);
        }

        /// <summary>
        /// Creates and then deletes non-versioned Amazon S3 objects and then
deletes
        /// them again. The method returns a list of the Amazon S3 objects
deleted.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// PubObjectsAsync and NonVersionedDeleteAsync.</param>
        /// <param name="bucketName">The name of the bucket where the objects
        /// will be created and then deleted.</param>
        /// <returns>A list of DeletedObjects.</returns>
        public static async Task<List<DeletedObject>>
DeleteObjectsAsync(IAmazonS3 client, string bucketName)
        {
            // Upload the sample objects.
            var keysAndVersions2 = await PutObjectsAsync(client, bucketName, 3);

            // Delete objects using only keys. Amazon S3 creates a delete marker
and
            // returns its version ID in the response.
            List<DeletedObject> deletedObjects = await
NonVersionedDeleteAsync(client, bucketName, keysAndVersions2);
            return deletedObjects;
        }

        /// <summary>
        /// This method creates several temporary objects and then deletes them.
        /// </summary>
        /// <param name="client">The S3 client.</param>
```

```
        /// <param name="bucketName">Name of the bucket.</param>
        /// <returns>Async task.</returns>
        public static async Task DeleteObjectVersionsAsync(IAmazonS3 client,
 string bucketName)
        {
            // Upload the sample objects.
            var keysAndVersions1 = await PutObjectsAsync(client, bucketName, 3);

            // Delete the specific object versions.
            await VersionedDeleteAsync(client, bucketName, keysAndVersions1);
        }

        /// <summary>
        /// Displays the list of information about deleted files to the console.
        /// </summary>
        /// <param name="e">Error information from the delete process.</param>
        private static void DisplayDeletionErrors(DeleteObjectsException e)
        {
            var errorResponse = e.Response;
            Console.WriteLine($"No. of objects successfully deleted =
 {errorResponse.DeletedObjects.Count}");
            Console.WriteLine($"No. of objects failed to delete =
 {errorResponse.DeleteErrors.Count}");
            Console.WriteLine("Printing error data...");
            foreach (var deleteError in errorResponse.DeleteErrors)
            {
                Console.WriteLine($"Object Key:
 {deleteError.Key}\t{deleteError.Code}\t{deleteError.Message}");
            }
        }

        /// <summary>
        /// Delete multiple objects from a version-enabled bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
 call
        /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
        /// RemoveDeleteMarkersAsync.</param>
        /// <param name="bucketName">The name of the bucket from which to delete
        /// objects.</param>
        /// <param name="keys">A list of key names for the objects to delete.</
param>
        private static async Task VersionedDeleteAsync(IAmazonS3 client, string
 bucketName, List<KeyVersion> keys)
```

```
        {
            var multiObjectDeleteRequest = new DeleteObjectsRequest
            {
                BucketName = bucketName,
                Objects = keys, // This includes the object keys and specific
 version IDs.
            };

            try
            {
                Console.WriteLine("Executing VersionedDelete...");
                DeleteObjectsResponse response = await
 client.DeleteObjectsAsync(multiObjectDeleteRequest);
                Console.WriteLine($"Successfully deleted all the
 {response.DeletedObjects.Count} items");
            }
            catch (DeleteObjectsException ex)
            {
                DisplayDeletionErrors(ex);
            }
        }

        /// <summary>
        /// Deletes multiple objects from a non-versioned Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
 call
        /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
        /// RemoveDeleteMarkersAsync.</param>
        /// <param name="bucketName">The name of the bucket from which to delete
        /// objects.</param>
        /// <param name="keys">A list of key names for the objects to delete.</
param>
        /// <returns>A list of the deleted objects.</returns>
        private static async Task<List<DeletedObject>>
 NonVersionedDeleteAsync(IAmazonS3 client, string bucketName, List<KeyVersion>
 keys)
        {
            // Create a request that includes only the object key names.
            DeleteObjectsRequest multiObjectDeleteRequest = new
 DeleteObjectsRequest();
            multiObjectDeleteRequest.BucketName = bucketName;

            foreach (var key in keys)
```

```
            {
                multiObjectDeleteRequest.AddKey(key.Key);
            }

            // Execute DeleteObjectsAsync.
            // The DeleteObjectsAsync method adds a delete marker for each
            // object deleted. You can verify that the objects were removed
            // using the Amazon S3 console.
            DeleteObjectsResponse response;
            try
            {
                Console.WriteLine("Executing NonVersionedDelete...");
                response = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
                Console.WriteLine("Successfully deleted all the {0} items",
response.DeletedObjects.Count);
            }
            catch (DeleteObjectsException ex)
            {
                DisplayDeletionErrors(ex);
                throw; // Some deletions failed. Investigate before continuing.
            }

            // This response contains the DeletedObjects list which we use to
delete the delete markers.
            return response.DeletedObjects;
        }

        /// <summary>
        /// Deletes the markers left after deleting the temporary objects.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// DeleteObjectVersionsAsync, DeleteObjectsAsync, and
        /// RemoveDeleteMarkersAsync.</param>
        /// <param name="bucketName">The name of the bucket from which to delete
        /// objects.</param>
        /// <param name="deletedObjects">A list of the objects that were
deleted.</param>
        private static async Task RemoveDeleteMarkersAsync(IAmazonS3 client,
string bucketName, List<DeletedObject> deletedObjects)
        {
            var keyVersionList = new List<KeyVersion>();
```

```
            foreach (var deletedObject in deletedObjects)
            {
                KeyVersion keyVersion = new KeyVersion
                {
                    Key = deletedObject.Key,
                    VersionId = deletedObject.DeleteMarkerVersionId,
                };
                keyVersionList.Add(keyVersion);
            }

            // Create another request to delete the delete markers.
            var multiObjectDeleteRequest = new DeleteObjectsRequest
            {
                BucketName = bucketName,
                Objects = keyVersionList,
            };

            // Now, delete the delete marker to bring your objects back to the
bucket.
            try
            {
                Console.WriteLine("Removing the delete markers .....");
                var deleteObjectResponse = await
client.DeleteObjectsAsync(multiObjectDeleteRequest);
                Console.WriteLine($"Successfully deleted the
{deleteObjectResponse.DeletedObjects.Count} delete markers");
            }
            catch (DeleteObjectsException ex)
            {
                DisplayDeletionErrors(ex);
            }
        }

        /// <summary>
        /// Create temporary Amazon S3 objects to show how object deletion wors
in an
        /// Amazon S3 bucket with versioning enabled.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// PutObjectAsync to create temporary objects for the example.</param>
        /// <param name="bucketName">A string representing the name of the S3
        /// bucket where we will create the temporary objects.</param>
```

```
        /// <param name="number">The number of temporary objects to create.</
param>
        /// <returns>A list of the KeyVersion objects.</returns>
        private static async Task<List<KeyVersion>> PutObjectsAsync(IAmazonS3
 client, string bucketName, int number)
        {
            var keys = new List<KeyVersion>();

            for (var i = 0; i < number; i++)
            {
                string key = "ObjectToDelete-" + new System.Random().Next();
                PutObjectRequest request = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = key,
                    ContentBody = "This is the content body!",
                };

                var response = await client.PutObjectAsync(request);
                KeyVersion keyVersion = new KeyVersion
                {
                    Key = key,
                    VersionId = response.VersionId,
                };

                keys.Add(keyVersion);
            }

            return keys;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
```

```
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n
$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::deleteObjects(const std::vector<Aws::String> &objectKeys,
                               const Aws::String &fromBucket,
                               const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::DeleteObjectsRequest request;

    Aws::S3::Model::Delete deleteObject;
    for (const Aws::String &objectKey: objectKeys) {

 deleteObject.AddObjects(Aws::S3::Model::ObjectIdentifier().WithKey(objectKey));
```

```
    }

    request.SetDelete(deleteObject);
    request.SetBucket(fromBucket);

    Aws::S3::Model::DeleteObjectsOutcome outcome =
            client.DeleteObjects(request);

    if (!outcome.IsSuccess()) {
        auto err = outcome.GetError();
        std::cerr << "Error deleting objects. " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Successfully deleted the objects.";
        for (size_t i = 0; i < objectKeys.size(); ++i) {
            std::cout << objectKeys[i];
            if (i < objectKeys.size() - 1) {
                std::cout << ", ";
            }
        }

        std::cout << " from bucket " << fromBucket << "." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante supprime un objet d'un compartiment nommé `my-bucket` :

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json`est un document JSON dans le répertoire en cours qui indique l'objet à supprimer :

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

Sortie :

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}
```

```
// DeleteObjects deletes a list of objects from a bucket.
func (actor S3Actions) DeleteObjects(ctx context.Context, bucket string, objects
 []types.ObjectIdentifier, bypassGovernance bool) error {
 if len(objects) == 0 {
  return nil
 }

 input := s3.DeleteObjectsInput{
  Bucket: aws.String(bucket),
  Delete: &types.Delete{
   Objects: objects,
   Quiet:   aws.Bool(true),
  },
 }
 if bypassGovernance {
  input.BypassGovernanceRetention = aws.Bool(true)
 }
 delOut, err := actor.S3Client.DeleteObjects(ctx, &input)
 if err != nil || len(delOut.Errors) > 0 {
  log.Printf("Error deleting objects from bucket %s.\n", bucket)
  if err != nil {
   var noBucket *types.NoSuchBucket
   if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
   }
  } else if len(delOut.Errors) > 0 {
   for _, outErr := range delOut.Errors {
    log.Printf("%s: %s\n", *outErr.Key, *outErr.Message)
   }
   err = fmt.Errorf("%s", *delOut.Errors[0].Message)
  }
 }
 return err
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.Delete;
import software.amazon.awssdk.services.s3.model.DeleteObjectsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class DeleteMultiObjects {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <bucketName>

                Where:
                   bucketName - the Amazon S3 bucket name.
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        deleteBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void deleteBucketObjects(S3Client s3, String bucketName) {
        // Upload three sample objects to the specfied Amazon S3 bucket.
        ArrayList<ObjectIdentifier> keys = new ArrayList<>();
        PutObjectRequest putOb;
        ObjectIdentifier objectId;

        for (int i = 0; i < 3; i++) {
            String keyName = "delete object example " + i;
            objectId = ObjectIdentifier.builder()
                    .key(keyName)
                    .build();

            putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(keyName)
                    .build();

            s3.putObject(putOb, RequestBody.fromString(keyName));
            keys.add(objectId);
        }

        System.out.println(keys.size() + " objects successfully created.");

        // Delete multiple objects in one request.
        Delete del = Delete.builder()
                .objects(keys)
                .build();

        try {
            DeleteObjectsRequest multiObjectDeleteRequest =
DeleteObjectsRequest.builder()
```

```
                    .bucket(bucketName)
                    .delete(del)
                    .build();

            s3.deleteObjects(multiObjectDeleteRequest);
            System.out.println("Multiple objects are deleted!");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez plusieurs objets.

```
import { DeleteObjectsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new DeleteObjectsCommand({
    Bucket: "test-bucket",
    Delete: {
      Objects: [{ Key: "object1.txt" }, { Key: "object2.txt" }],
    },
  });
```

```
  try {
    const { Deleted } = await client.send(command);
    console.log(
      `Successfully deleted ${Deleted.length} objects from S3 bucket. Deleted
 objects:`,
    );
    console.log(Deleted.map((d) => ` • ${d.Key}`).join("\n"));
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deleteBucketObjects(
    bucketName: String,
    objectName: String,
) {
    val objectId =
        ObjectIdentifier {
            key = objectName
        }

    val delOb =
        Delete {
            objects = listOf(objectId)
        }
```

```
    val request =
        DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteObjects(request)
        println("$objectName was deleted from $bucketName")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un ensemble d'objets d'une liste de clés.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $objects = [];
            foreach ($contents['Contents'] as $content) {
                $objects[] = [
                    'Key' => $content['Key'],
                ];
            }
            $this->s3client->deleteObjects([
                'Bucket' => $this->bucketName,
                'Delete' => [
                    'Objects' => $objects,
                ],
```

```
            ]);
            $check = $this->s3client->listObjectsV2([
                'Bucket' => $this->bucketName,
            ]);
            if (count($check) <= 0) {
                throw new Exception("Bucket wasn't empty.");
            }
            echo "Deleted all objects and folders from $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to delete $fileName from $this->bucketName with error:
" . $exception->getMessage();
            exit("Please fix error with object deletion before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for PHP API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande supprime l'objet « sample.txt » du bucket « test-files ». Vous êtes invité à confirmer avant l'exécution de la commande ; pour supprimer l'invite, utilisez le commutateur -Force.

```
Remove-S3Object -BucketName test-files -Key sample.txt
```

Exemple 2 : Cette commande supprime la version spécifiée de l'objet « sample.txt » du bucket « test-files », en supposant que le bucket a été configuré pour activer les versions de l'objet.

```
Remove-S3Object -BucketName test-files -Key sample.txt -VersionId
 HLbxnx6V9omT6AQYVpks8mmFKQcejpqt
```

Exemple 3 : cette commande supprime les objets « sample1.txt », « sample2.txt » et « sample3.txt » du bucket « test-files » en une seule opération par lots. La réponse du service listera toutes les clés traitées, quel que soit le statut de réussite ou d'erreur de la suppression. Pour obtenir uniquement les erreurs pour les clés qui n'ont pas pu être traitées par le service, ajoutez le ReportErrorsOnly paramètre - (ce paramètre peut également être spécifié avec l'alias -Quiet).

```
Remove-S3Object -BucketName test-files -KeyCollection @( "sample1.txt",
 "sample2.txt", "sample3.txt" )
```

Exemple 4 : Cet exemple utilise une expression en ligne avec le KeyCollection paramètre - pour obtenir les clés des objets à supprimer. Get-S3Objectrenvoie une collection d'instances Amazon.S3.Model.S3Object, dont chacune possède un membre clé de type chaîne identifiant l'objet.

```
Remove-S3Object -bucketname "test-files" -KeyCollection (Get-S3Object "test-
files" -KeyPrefix "prefix/subprefix" | select -ExpandProperty Key)
```

Exemple 5 : Cet exemple obtient tous les objets dont le préfixe clé est « préfixe/sous-préfixe » dans le compartiment et les supprime. Notez que les objets entrants sont traités un par un. Pour les collections volumineuses, pensez à transmettre la collection au paramètre - InputObject (alias -S3ObjectCollection) de l'applet de commande pour permettre à la suppression de se produire par lots avec un seul appel au service.

```
Get-S3Object -BucketName "test-files" -KeyPrefix "prefix/subprefix" | Remove-
S3Object -Force
```

Exemple 6 : Cet exemple dirige une collection d'ObjectVersion instances Amazon.S3.Model.S3 qui représentent des marqueurs de suppression vers l'applet de commande pour suppression. Notez que les objets entrants sont traités un par un. Pour les collections volumineuses, pensez à transmettre la collection au paramètre - InputObject (alias -S3ObjectCollection) de l'applet de commande pour permettre à la suppression de se produire par lots avec un seul appel au service.

```
(Get-S3Version -BucketName "test-files").Versions | Where {$_.IsDeleteMarker -eq
 "True"} | Remove-S3Object -Force
```

Exemple 7 : Ce script montre comment supprimer par lots un ensemble d'objets (dans ce cas, des marqueurs de suppression) en créant un tableau d'objets à utiliser avec le KeyAndVersionCollection paramètre -.

```
$keyVersions = @()
$markers = (Get-S3Version -BucketName $BucketName).Versions | Where
 {$_.IsDeleteMarker -eq "True"}
```

```
foreach ($marker in $markers) { $keyVersions += @{ Key = $marker.Key; VersionId =
 $marker.VersionId } }
Remove-S3Object -BucketName $BucketName -KeyAndVersionCollection $keyVersions -
Force
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez un ensemble d'objets à l'aide d'une liste de clés d'objet.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    @staticmethod
    def delete_objects(bucket, object_keys):
        """
        Removes a list of objects from a bucket.
        This operation is done as a batch in a single request.

        :param bucket: The bucket that contains the objects. This is a Boto3
 Bucket
                       resource.
```

```
        :param object_keys: The list of keys that identify the objects to remove.
        :return: The response that contains data about which objects were deleted
                 and any that could not be deleted.
        """
        try:
            response = bucket.delete_objects(
                Delete={"Objects": [{"Key": key} for key in object_keys]}
            )
            if "Deleted" in response:
                logger.info(
                    "Deleted objects '%s' from bucket '%s'.",
                    [del_obj["Key"] for del_obj in response["Deleted"]],
                    bucket.name,
                )
            if "Errors" in response:
                logger.warning(
                    "Could not delete objects '%s' from bucket '%s'.",
                    [
                        f"{del_obj['Key']}: {del_obj['Code']}"
                        for del_obj in response["Errors"]
                    ],
                    bucket.name,
                )
        except ClientError:
            logger.exception("Couldn't delete any objects from bucket %s.",
 bucket.name)
            raise
        else:
            return response
```

Supprimez tous les objets dans un compartiment.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
```

```
        self.object = s3_object
        self.key = self.object.key


    @staticmethod
    def empty_bucket(bucket):
        """
        Remove all objects from a bucket.

        :param bucket: The bucket to empty. This is a Boto3 Bucket resource.
        """
        try:
            bucket.objects.delete()
            logger.info("Emptied bucket '%s'.", bucket.name)
        except ClientError:
            logger.exception("Couldn't empty bucket '%s'.", bucket.name)
            raise
```

Supprimez définitivement un objet soumis à la gestion des versions en supprimant toutes ses versions.

```
def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

- Pour plus de détails sur l'API, consultez DeleteObjectsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
 # Deletes the objects in an Amazon S3 bucket and deletes the bucket.
 #
 # @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
 def delete_bucket(bucket)
   puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?
")
   answer = gets.chomp.downcase
   if answer == "y"
     bucket.objects.batch_delete!
     bucket.delete
     puts("Emptied and deleted bucket #{bucket.name}.\n")
   end
 rescue Aws::Errors::ServiceError => e
   puts("Couldn't empty and delete bucket #{bucket.name}.")
   puts("\t#{e.code}: #{e.message}")
   raise
 end
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn delete_objects(client: &Client, bucket_name: &str) ->
 Result<Vec<String>, Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    let mut delete_objects: Vec<ObjectIdentifier> = vec![];
    for obj in objects.contents() {
        let obj_id = ObjectIdentifier::builder()
            .set_key(Some(obj.key().unwrap().to_string()))
            .build()
            .map_err(Error::from)?;
        delete_objects.push(obj_id);
    }

    let return_keys = delete_objects.iter().map(|o| o.key.clone()).collect();

    if !delete_objects.is_empty() {
        client
            .delete_objects()
            .bucket(bucket_name)
            .delete(
                Delete::builder()
                    .set_objects(Some(delete_objects))
                    .build()
                    .map_err(Error::from)?,
            )
            .send()
            .await?;
    }

    let objects: ListObjectsV2Output =
 client.list_objects_v2().bucket(bucket_name).send().await?;
```

```
    eprintln!("{objects:?}");

    match objects.key_count {
        Some(0) => Ok(return_keys),
        _ => Err(Error::unhandled(
            "There were still objects left in the bucket.",
        )),
    }
}
```

- Pour plus de détails sur l'API, voir DeleteObjectsla section de référence de l'API AWS SDK for Rust.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public func deleteObjects(bucket: String, keys: [String]) async throws {
        let input = DeleteObjectsInput(
            bucket: bucket,
            delete: S3ClientTypes.Delete(
                objects: keys.map({ S3ClientTypes.ObjectIdentifier(key: $0) }),
                quiet: true
            )
        )

        do {
```

```
            let output = try await client.deleteObjects(input: input)

            // As of the last update to this example, any errors are returned
            // in the `output` object's `errors` property. If there are any
            // errors in this array, throw an exception. Once the error
            // handling is finalized in later updates to the AWS SDK for
            // Swift, this example will be updated to handle errors better.

            guard let errors = output.errors else {
                return  // No errors.
            }
            if errors.count != 0 {
                throw ServiceHandlerError.deleteObjectsError
            }
        } catch {
            throw error
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteObjectsà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeletePublicAccessBlock`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`DeletePublicAccessBlock`.

CLI

AWS CLI

Pour supprimer la configuration de blocage de l'accès public pour un bucket

L'`delete-public-access-block`exemple suivant supprime la configuration de blocage de l'accès public sur le compartiment spécifié.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous DeletePublicAccessBlockà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande désactive le paramètre de blocage de l'accès public pour le bucket donné.

```
Remove-S3PublicAccessBlock -BucketName 's3testbucket' -Force -Select
 '^BucketName'
```

Sortie :

```
s3testbucket
```

- Pour plus de détails sur l'API, reportez-vous DeletePublicAccessBlockà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketAccelerateConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketAccelerateConfiguration`.

CLI

AWS CLI

Pour récupérer la configuration accélérée d'un bucket

L'`get-bucket-accelerate-configuration`exemple suivant récupère la configuration d'accélération pour le compartiment spécifié.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket my-bucket
```

Sortie :

```
{
    "Status": "Enabled"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAccelerateConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie la valeur Enabled, si les paramètres d'accélération du transfert sont activés pour le compartiment spécifié.

```
Get-S3BucketAccelerateConfiguration -BucketName 's3testbucket'
```

Sortie :

```
Value
-----
Enabled
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAccelerateConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketAcl`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketAcl`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Gérer les listes de contrôle d'accès (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Get the access control list (ACL) for the new bucket.
        /// </summary>
        /// <param name="client">The initialized client object used to get the
        /// access control list (ACL) of the bucket.</param>
        /// <param name="newBucketName">The name of the newly created bucket.</
param>
        /// <returns>An S3AccessControlList.</returns>
        public static async Task<S3AccessControlList>
 GetACLForBucketAsync(IAmazonS3 client, string newBucketName)
        {
            // Retrieve bucket ACL to show that the ACL was properly applied to
            // the new bucket.
            GetACLResponse getACLResponse = await client.GetACLAsync(new
 GetACLRequest
            {
                BucketName = newBucketName,
            });

            return getACLResponse.AccessControlList;
        }
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::getBucketAcl(const Aws::String &bucketName,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetBucketAclRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketAclOutcome outcome =
            s3Client.GetBucketAcl(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getBucketAcl: "
                  << err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        Aws::Vector<Aws::S3::Model::Grant> grants =
                outcome.GetResult().GetGrants();

        for (auto it = grants.begin(); it != grants.end(); it++) {
            Aws::S3::Model::Grant grant = *it;
            Aws::S3::Model::Grantee grantee = grant.GetGrantee();

            std::cout << "For bucket " << bucketName << ": "
                      << std::endl << std::endl;

            if (grantee.TypeHasBeenSet()) {
```

```
                std::cout << "Type:          "
                          << getGranteeTypeString(grantee.GetType()) <<
 std::endl;
            }

            if (grantee.DisplayNameHasBeenSet()) {
                std::cout << "Display name:  "
                          << grantee.GetDisplayName() << std::endl;
            }

            if (grantee.EmailAddressHasBeenSet()) {
                std::cout << "Email address: "
                          << grantee.GetEmailAddress() << std::endl;
            }

            if (grantee.IDHasBeenSet()) {
                std::cout << "ID:            "
                          << grantee.GetID() << std::endl;
            }

            if (grantee.URIHasBeenSet()) {
                std::cout << "URI:           "
                          << grantee.GetURI() << std::endl;
            }

            std::cout << "Permission:    " <<
                      getPermissionString(grant.GetPermission()) <<
                      std::endl << std::endl;
        }
    }

    return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
 string.
/*!
 \param type: Type enumeration.
 \return String: Human-readable string.
*/

Aws::String getGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {
        case Aws::S3::Model::Type::AmazonCustomerByEmail:
```

```
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:
            return "Canonical user ID of an AWS account";
        case Aws::S3::Model::Type::Group:
            return "Predefined Amazon S3 group";
        case Aws::S3::Model::Type::NOT_SET:
            return "Not set";
        default:
            return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
 string.
/*!
 \param permission: Permission enumeration.
 \return String: Human-readable string.
*/

Aws::String getPermissionString(const Aws::S3::Model::Permission &permission) {
    switch (permission) {
        case Aws::S3::Model::Permission::FULL_CONTROL:
            return "Can list objects in this bucket, create/overwrite/delete "
                   "objects in this bucket, and read/write this "
                   "bucket's permissions";
        case Aws::S3::Model::Permission::NOT_SET:
            return "Permission not set";
        case Aws::S3::Model::Permission::READ:
            return "Can list objects in this bucket";
        case Aws::S3::Model::Permission::READ_ACP:
            return "Can read this bucket's permissions";
        case Aws::S3::Model::Permission::WRITE:
            return "Can create, overwrite, and delete objects in this bucket";
        case Aws::S3::Model::Permission::WRITE_ACP:
            return "Can write this bucket's permissions";
        default:
            return "Permission unknown";
    }

    return "Permission unknown";
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante permet de récupérer la liste de contrôle d'accès pour un bucket nommé `my-bucket` :

```
aws s3api get-bucket-acl --bucket my-bucket
```

Sortie :

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectAclRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAclResponse;
import software.amazon.awssdk.services.s3.model.Grant;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetAcl {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <bucketName> <objectKey>

                Where:
                  bucketName - The Amazon S3 bucket to get the access control
 list (ACL) for.
                  objectKey - The object to get the ACL for.\s
                """;

        if (args.length != 2) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        System.out.println("Retrieving ACL for object: " + objectKey);
        System.out.println("in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getBucketACL(s3, objectKey, bucketName);
        s3.close();
        System.out.println("Done!");
    }

    public static String getBucketACL(S3Client s3, String objectKey, String
 bucketName) {
        try {
            GetObjectAclRequest aclReq = GetObjectAclRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();

            GetObjectAclResponse aclRes = s3.getObjectAcl(aclReq);
            List<Grant> grants = aclRes.grants();
            String grantee = "";
            for (Grant grant : grants) {
                System.out.format("  %s: %s\n", grant.grantee().id(),
 grant.permission());
                grantee = grant.grantee().id();
            }

            return grantee;
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return "";
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez les autorisations ACL.

```
import { GetBucketAclCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketAclCommand({
    Bucket: "test-bucket",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous GetBucketAclà la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def get_acl(self):
        """
        Get the ACL of the bucket.

        :return: The ACL of the bucket.
        """
        try:
            acl = self.bucket.Acl()
            logger.info(
                "Got ACL for bucket %s. Owner is %s.", self.bucket.name,
 acl.owner
            )
        except ClientError:
            logger.exception("Couldn't get ACL for bucket %s.", self.bucket.name)
            raise
        else:
            return acl
```

- Pour plus de détails sur l'API, consultez GetBucketAcl le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetBucketAnalyticsConfiguration** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketAnalyticsConfiguration`.

CLI

AWS CLI

Pour récupérer la configuration d'analyse d'un bucket avec un ID spécifique

L'`get-bucket-analytics-configuration`exemple suivant affiche la configuration d'analyse pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

Sortie :

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAnalyticsConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les détails du filtre d'analyse nommé « testfilter » dans le compartiment S3 donné.

```
Get-S3BucketAnalyticsConfiguration -BucketName 's3testbucket' -AnalyticsId
 'testfilter'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketAnalyticsConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketCors`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Retrieve the CORS configuration applied to the Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used
        /// to retrieve the CORS configuration.</param>
        /// <returns>The created CORS configuration object.</returns>
        private static async Task<CORSConfiguration>
RetrieveCORSConfigurationAsync(AmazonS3Client client)
        {
```

```
            GetCORSConfigurationRequest request = new
GetCORSConfigurationRequest()
            {
                BucketName = BucketName,
            };
            var response = await client.GetCORSConfigurationAsync(request);
            var configuration = response.Configuration;
            PrintCORSRules(configuration);
            return configuration;
        }
```

- Pour plus de détails sur l'API, reportez-vous GetBucketCorsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante extrait la configuration du partage de ressources entre origines pour un compartiment nommé : `my-bucket`

```
aws s3api get-bucket-cors --bucket my-bucket
```

Sortie :

```
{
    "CORSRules": [
        {
            "AllowedHeaders": [
                "*"
            ],
            "ExposeHeaders": [
                "x-amz-server-side-encryption"
            ],
            "AllowedMethods": [
                "PUT",
                "POST",
                "DELETE"
            ],
            "MaxAgeSeconds": 3000,
```

```
            "AllowedOrigins": [
                "http://www.example.com"
            ]
        },
        {
            "AllowedHeaders": [
                "Authorization"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedMethods": [
                "GET"
            ],
            "AllowedOrigins": [
                "*"
            ]
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketCorsà la section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez la politique CORS pour le compartiment.

```
import { GetBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketCorsCommand({
    Bucket: "test-bucket",
  });
```

```
  try {
    const { CORSRules } = await client.send(command);
    CORSRules.forEach((cr, i) => {
      console.log(
        `\nCORSRule ${i + 1}`,
        `\n${"-".repeat(10)}`,
        `\nAllowedHeaders: ${cr.AllowedHeaders.join(" ")}`,
        `\nAllowedMethods: ${cr.AllowedMethods.join(" ")}`,
        `\nAllowedOrigins: ${cr.AllowedOrigins.join(" ")}`,
        `\nExposeHeaders: ${cr.ExposeHeaders.join(" ")}`,
        `\nMaxAgeSeconds: ${cr.MaxAgeSeconds}`,
      );
    });
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous GetBucketCorsà la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
```

```
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def get_cors(self):
        """
        Get the CORS rules for the bucket.

        :return The CORS rules for the specified bucket.
        """
        try:
            cors = self.bucket.Cors()
            logger.info(
                "Got CORS rules %s for bucket '%s'.", cors.cors_rules,
self.bucket.name
            )
        except ClientError:
            logger.exception(("Couldn't get CORS for bucket %s.",
self.bucket.name))
            raise
        else:
            return cors
```

- Pour plus de détails sur l'API, consultez GetBucketCorsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
 an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Gets the CORS configuration of a bucket.
  #
  # @return [Aws::S3::Type::GetBucketCorsOutput, nil] The current CORS
 configuration for the bucket.
  def get_cors
    @bucket_cors.data
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get CORS configuration for #{@bucket_cors.bucket.name}. Here's
 why: #{e.message}"
    nil
  end

end
```

- Pour plus de détails sur l'API, reportez-vous GetBucketCorsà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketEncryption`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketEncryption`.

CLI

AWS CLI

Pour récupérer la configuration de chiffrement côté serveur pour un bucket

L'`get-bucket-encryption`exemple suivant récupère la configuration de chiffrement côté serveur pour le compartiment. `my-bucket`

```
aws s3api get-bucket-encryption \
    --bucket my-bucket
```

Sortie :

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketEncryptionà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie toutes les règles de chiffrement côté serveur associées au bucket donné.

```
Get-S3BucketEncryption -BucketName 's3casetestbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketEncryptionà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetBucketInventoryConfiguration** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketInventoryConfiguration`.

CLI

AWS CLI

Pour récupérer la configuration d'inventaire d'un bucket

L'`get-bucket-inventory-configuration`exemple suivant récupère la configuration d'inventaire pour le compartiment spécifié avec l'ID1.

```
aws s3api get-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

Sortie :

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::my-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketInventoryConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les détails de l'inventaire nommé « testinventory » pour le compartiment S3 donné.

```
Get-S3BucketInventoryConfiguration -BucketName 's3testbucket' -InventoryId
 'testinventory'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketInventoryConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketLifecycleConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketLifecycleConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Returns a configuration object for the supplied bucket name.
        /// </summary>
```

```
        /// <param name="client">The S3 client object used to call
        /// the GetLifecycleConfigurationAsync method.</param>
        /// <param name="bucketName">The name of the S3 bucket for which a
        /// configuration will be created.</param>
        /// <returns>Returns a new LifecycleConfiguration object.</returns>
        public static async Task<LifecycleConfiguration>
 RetrieveLifecycleConfigAsync(IAmazonS3 client, string bucketName)
        {
            var request = new GetLifecycleConfigurationRequest()
            {
                BucketName = bucketName,
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLifecycleConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration du cycle de vie d'un compartiment nommé `my-bucket` :

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

Sortie :

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
```

```
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
                    "NoncurrentDays": 0,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLifecycleConfigurationà la section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
```

```
        self.name = bucket.name


    def get_lifecycle_configuration(self):
        """
        Get the lifecycle configuration of the bucket.

        :return: The lifecycle rules of the specified bucket.
        """
        try:
            config = self.bucket.LifecycleConfiguration()
            logger.info(
                "Got lifecycle rules %s for bucket '%s'.",
                config.rules,
                self.bucket.name,
            )
        except:
            logger.exception(
                "Couldn't get lifecycle rules for bucket '%s'.", self.bucket.name
            )
            raise
        else:
            return config.rules
```

- Pour plus de détails sur l'API, consultez GetBucketLifecycleConfigurationle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketLocation`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketLocation`.

CLI

AWS CLI

La commande suivante récupère la contrainte d'emplacement pour un compartiment nommé`my-bucket`, s'il en existe une :

```
aws s3api get-bucket-location --bucket my-bucket
```

Sortie :

```
{
    "LocationConstraint": "us-west-2"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLocationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie la contrainte d'emplacement pour le bucket « s3testbucket », s'il existe une contrainte.

```
Get-S3BucketLocation -BucketName 's3testbucket'
```

Sortie :

```
Value
-----
ap-south-1
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLocationà la section Référence des AWS Tools for PowerShell applets de commande.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(),
 Error> {
    let resp = client.list_buckets().send().await?;
    let buckets = resp.buckets();
    let num_buckets = buckets.len();

    let mut in_region = 0;

    for bucket in buckets {
        if strict {
            let r = client
                .get_bucket_location()
                .bucket(bucket.name().unwrap_or_default())
                .send()
                .await?;

            if r.location_constraint().unwrap().as_ref() == region {
                println!("{}", bucket.name().unwrap_or_default());
                in_region += 1;
            }
        } else {
            println!("{}", bucket.name().unwrap_or_default());
        }
    }

    println!();
    if strict {
        println!(
            "Found {} buckets in the {} region out of a total of {} buckets.",
            in_region, region, num_buckets
        );
    } else {
```

```
        println!("Found {} buckets in all regions.", num_buckets);
    }

    Ok(())
}
```

- Pour plus de détails sur l'API, voir GetBucketLocationla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketLogging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketLogging`.

CLI

AWS CLI

Pour récupérer l'état de journalisation d'un bucket

L'`get-bucket-logging`exemple suivant permet de récupérer l'état de journalisation pour le compartiment spécifié.

```
aws s3api get-bucket-logging \
    --bucket my-bucket
```

Sortie :

```
{
    "LoggingEnabled": {
        "TargetPrefix": "",
        "TargetBucket": "my-bucket-logs"
          }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLoggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie l'état de journalisation pour le compartiment spécifié.

```
Get-S3BucketLogging -BucketName 's3testbucket'
```

Sortie :

```
TargetBucketName   Grants TargetPrefix
----------------   ------ ------------
testbucket1        {}     testprefix
```

- Pour plus de détails sur l'API, reportez-vous GetBucketLoggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketMetricsConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketMetricsConfiguration`.

CLI

AWS CLI

Pour récupérer la configuration des métriques d'un bucket avec un ID spécifique

L'`get-bucket-metrics-configuration`exemple suivant affiche la configuration des métriques pour le bucket et l'ID spécifiés.

```
aws s3api get-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

Sortie :

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketMetricsConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les détails du filtre de métriques nommé « testfilter » pour le compartiment S3 donné.

```
Get-S3BucketMetricsConfiguration -BucketName 's3testbucket' -MetricsId
 'testfilter'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketMetricsConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketNotification`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketNotification`.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration des notifications pour un compartiment nommé `my-bucket` :

```
aws s3api get-bucket-notification --bucket my-bucket
```

Sortie :

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketNotificationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cet exemple récupère la configuration des notifications du bucket donné

```
Get-S3BucketNotification -BucketName kt-tools | select -ExpandProperty
 TopicConfigurations
```

Sortie :

```
Id   Topic
--   -----
mimo arn:aws:sns:eu-west-1:123456789012:topic-1
```

- Pour plus de détails sur l'API, reportez-vous GetBucketNotificationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketPolicy`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketPolicy`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::getBucketPolicy(const Aws::String &bucketName,
                                 const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetBucketPolicyRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketPolicyOutcome outcome =
            s3Client.GetBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getBucketPolicy: "
                  << err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        Aws::StringStream policy_stream;
        Aws::String line;

        outcome.GetResult().GetPolicy() >> line;
        policy_stream << line;

        std::cout << "Retrieve the policy for bucket '" << bucketName << "':\n\n"
 <<
                  policy_stream.str() << std::endl;
    }
```

```
    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante permet de récupérer la politique de compartiment pour un compartiment nommé `my-bucket` :

```
aws s3api get-bucket-policy --bucket my-bucket
```

Sortie :

```
{
    "Policy": "{\"Version\":\"2008-10-17\",\"Statement\":[{\"Sid\":\"\",\"Effect
\":\"Allow\",\"Principal\":\"*\",\"Action\":\"s3:GetObject\",\"Resource\":
\"arn:aws:s3:::my-bucket/*\"},{\"Sid\":\"\",\"Effect\":\"Deny\",\"Principal\":
\"*\",\"Action\":\"s3:GetObject\",\"Resource\":\"arn:aws:s3:::my-bucket/secret/*
\"}]}"
}
```

Obtenir et mettre une politique de compartiment L'exemple suivant montre comment télécharger une politique de compartiment Amazon S3, apporter des modifications au fichier, puis utiliser `put-bucket-policy` pour appliquer la politique de compartiment modifiée. Pour télécharger la politique du bucket dans un fichier, vous pouvez exécuter :

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
 policy.json
```

Vous pouvez ensuite modifier le `policy.json` fichier selon vos besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

`policy.json`fichier selon les besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

fichier selon les besoins. Enfin, vous pouvez réappliquer cette politique modifiée au compartiment S3 en exécutant :

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetBucketPolicyRequest;
import software.amazon.awssdk.services.s3.model.GetBucketPolicyResponse;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetBucketPolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>
```

```
                Where:
                    bucketName - The Amazon S3 bucket to get the policy from.
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        String polText = getPolicy(s3, bucketName);
        System.out.println("Policy Text: " + polText);
        s3.close();
    }

    public static String getPolicy(S3Client s3, String bucketName) {
        String policyText;
        System.out.format("Getting policy for bucket: \"%s\"\n\n", bucketName);
        GetBucketPolicyRequest policyReq = GetBucketPolicyRequest.builder()
                .bucket(bucketName)
                .build();

        try {
            GetBucketPolicyResponse policyRes = s3.getBucketPolicy(policyReq);
            policyText = policyRes.policy();
            return policyText;

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return "";
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez la politique de compartiment.

```
import { GetBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketPolicyCommand({
    Bucket: "test-bucket",
  });

  try {
    const { Policy } = await client.send(command);
    console.log(JSON.parse(Policy));
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun getPolicy(bucketName: String): String? {
    println("Getting policy for bucket $bucketName")

    val request =
        GetBucketPolicyRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val policyRes = s3.getBucketPolicy(request)
        return policyRes.policy
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande génère la politique de compartiment associée au compartiment S3 donné.

```
Get-S3BucketPolicy -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def get_policy(self):
        """
        Get the security policy of the bucket.

        :return: The security policy of the specified bucket, in JSON format.
        """
        try:
            policy = self.bucket.Policy()
            logger.info(
                "Got policy %s for bucket '%s'.", policy.policy, self.bucket.name
            )
        except ClientError:
            logger.exception("Couldn't get policy for bucket '%s'.",
 self.bucket.name)
            raise
        else:
            return json.loads(policy.policy)
```

- Pour plus de détails sur l'API, consultez GetBucketPolicyle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
 configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Gets the policy of a bucket.
  #
  # @return [Aws::S3::GetBucketPolicyOutput, nil] The current bucket policy.
  def get_policy
    policy = @bucket_policy.data.policy
    policy.respond_to?(:read) ? policy.read : policy
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get the policy for #{@bucket_policy.bucket.name}. Here's why:
 #{e.message}"
    nil
  end

end
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetBucketPolicyStatus** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliserGetBucketPolicyStatus.

CLI

AWS CLI

Pour récupérer le statut de la politique d'un compartiment indiquant s'il est public

L'get-bucket-policy-statusexemple suivant permet de récupérer le statut de la politique pour le compartimentmy-bucket.

```
aws s3api get-bucket-policy-status \
    --bucket my-bucket
```

Sortie :

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyStatusà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie l'état de la politique pour le compartiment S3 donné, indiquant si le compartiment est public.

```
Get-S3BucketPolicyStatus -BucketName 's3casetestbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketPolicyStatusà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketReplication`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketReplication`.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration de réplication pour un compartiment nommé `my-bucket` :

```
aws s3api get-bucket-replication --bucket my-bucket
```

Sortie :

```
{
    "ReplicationConfiguration": {
        "Rules": [
            {
                "Status": "Enabled",
                "Prefix": "",
                "Destination": {
                    "Bucket": "arn:aws:s3:::my-bucket-backup",
                    "StorageClass": "STANDARD"
                },
                "ID": "ZmUwNzE4ZmQ4tMjVhOS00MTlkLOGI4NDkzZTIWJjNTUtYTA1"
            }
        ],
        "Role": "arn:aws:iam::123456789012:role/s3-replication-role"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketReplicationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : renvoie les informations de configuration de réplication définies sur le compartiment nommé « mybucket ».

```
Get-S3BucketReplication -BucketName mybucket
```

- Pour plus de détails sur l'API, reportez-vous GetBucketReplicationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketRequestPayment`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketRequestPayment`.

CLI

AWS CLI

Pour récupérer la configuration de paiement de la demande pour un bucket

L'`get-bucket-request-payment`exemple suivant extrait la configuration du paiement par le demandeur pour le compartiment spécifié.

```
aws s3api get-bucket-request-payment \
    --bucket my-bucket
```

Sortie :

```
{
    "Payer": "BucketOwner"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketRequestPaymentà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : renvoie la configuration de paiement de la demande pour le compartiment nommé « mybucket ». Par défaut, le propriétaire du bucket paie les téléchargements depuis le bucket.

```
Get-S3BucketRequestPayment -BucketName mybucket
```

- Pour plus de détails sur l'API, reportez-vous GetBucketRequestPaymentà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketTagging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketTagging`.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration de balisage pour un compartiment nommé : `my-bucket`

```
aws s3api get-bucket-tagging --bucket my-bucket
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketTaggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie toutes les balises associées au bucket donné.

```
Get-S3BucketTagging -BucketName 's3casetestbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketTaggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketVersioning`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketVersioning`.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration de version pour un compartiment nommé : `my-bucket`

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Sortie :

```
{
    "Status": "Enabled"
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketVersioningà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie l'état du versionnement par rapport au bucket donné.

```
Get-S3BucketVersioning -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketVersioningà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetBucketWebsite`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
                // Get the website configuration.
                GetBucketWebsiteRequest getRequest = new
GetBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                };
                GetBucketWebsiteResponse getResponse = await
client.GetBucketWebsiteAsync(getRequest);
                Console.WriteLine($"Index document:
{getResponse.WebsiteConfiguration.IndexDocumentSuffix}");
                Console.WriteLine($"Error document:
{getResponse.WebsiteConfiguration.ErrorDocument}");
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::getWebsiteConfig(const Aws::String &bucketName,
                                  const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetBucketWebsiteRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::GetBucketWebsiteOutcome outcome =
            s3Client.GetBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();

        std::cerr << "Error: GetBucketWebsite: "
                  << err.GetMessage() << std::endl;
    } else {
        Aws::S3::Model::GetBucketWebsiteResult websiteResult =
 outcome.GetResult();

        std::cout << "Success: GetBucketWebsite: "
                  << std::endl << std::endl
                  << "For bucket '" << bucketName << "':"
                  << std::endl
                  << "Index page : "
```

```
                  << websiteResult.GetIndexDocument().GetSuffix()
                  << std::endl
                  << "Error page: "
                  << websiteResult.GetErrorDocument().GetKey()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante permet de récupérer la configuration statique du site Web pour un compartiment nommé `my-bucket` :

```
aws s3api get-bucket-website --bucket my-bucket
```

Sortie :

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS CLI commandes.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez la configuration du site web.

```
import { GetBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetBucketWebsiteCommand({
    Bucket: "test-bucket",
  });

  try {
    const { ErrorDocument, IndexDocument } = await client.send(command);
    console.log(
      `Your bucket is set up to host a website. It has an error document:`,
      `${ErrorDocument.Key}, and an index document: ${IndexDocument.Suffix}.`,
    );
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les détails des configurations de site Web statiques du compartiment S3 donné.

```
Get-S3BucketWebsite -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetBucketWebsiteà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObject`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Récupérer un objet d'un compartiment s'il a été modifié
- Obtenir un objet depuis un point d'accès multirégional
- Démarrer avec les compartiments et les objets
- Démarrer avec le chiffrement
- Suivez les chargements et les téléchargements

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Shows how to download an object from an Amazon S3 bucket to the
        /// local computer.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the bucket where the object is
        /// currently stored.</param>
        /// <param name="objectName">The name of the object to download.</param>
        /// <param name="filePath">The path, including filename, where the
        /// downloaded object will be stored.</param>
        /// <returns>A boolean value indicating the success or failure of the
        /// download process.</returns>
        public static async Task<bool> DownloadObjectFromBucketAsync(
            IAmazonS3 client,
            string bucketName,
            string objectName,
            string filePath)
        {
            // Create a GetObject request
            var request = new GetObjectRequest
            {
                BucketName = bucketName,
                Key = objectName,
            };

            // Issue request and remember to dispose of the response
            using GetObjectResponse response = await
 client.GetObjectAsync(request);

            try
            {
                // Save object to local file
                await response.WriteResponseStreamToFileAsync($"{filePath}\
\{objectName}", true, CancellationToken.None);
                return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error saving {objectName}: {ex.Message}");
                return false;
            }
        }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function download_object_from_bucket() {
  local bucket_name=$1
```

```
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::getObject(const Aws::String &objectKey,
                           const Aws::String &fromBucket,
                           const Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::GetObjectRequest request;
    request.SetBucket(fromBucket);
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectOutcome outcome =
            client.GetObject(request);
```

```
    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getObject: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Successfully retrieved '" << objectKey << "' from '"
                  << fromBucket << "'." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

L'exemple suivant utilise la `get-object` commande pour télécharger un objet depuis Amazon S3 :

```
aws s3api get-object --bucket text-content --key dir/my_images.tar.bz2
 my_images.tar.bz2
```

Notez que le paramètre outfile est spécifié sans nom d'option tel que « --outfile ». Le nom du fichier de sortie doit être le dernier paramètre de la commande.

L'exemple ci-dessous illustre l'utilisation de `--range` pour télécharger une plage d'octets spécifique à partir d'un objet. Notez que les plages d'octets doivent être préfixées par « bytes= » :

```
aws s3api get-object --bucket text-content --key dir/my_data --range
 bytes=8888-9999 my_data_range
```

Pour plus d'informations sur la récupération d'objets, consultez Getting Objects dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
 fileName string) error {
 result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
 objectKey, err)
  return err
 }
 defer result.Body.Close()
 file, err := os.Create(fileName)
 if err != nil {
```

```
  log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
  return err
 }
 defer file.Close()
 body, err := io.ReadAll(result.Body)
 if err != nil {
  log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
 err)
 }
 _, err = file.Write(body)
 return err
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Lisez des données sous forme de tableau d'octets en utilisant un S3Client.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
```

```
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetObjectData {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName> <path>

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - The key name.\s
                    path - The path where the file is written to.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String path = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getObjectBytes(s3, bucketName, keyName, path);
    }

    public static void getObjectBytes(S3Client s3, String bucketName, String
 keyName, String path) {
        try {
            GetObjectRequest objectRequest = GetObjectRequest
                    .builder()
                    .key(keyName)
```

```
                    .bucket(bucketName)
                    .build();

            ResponseBytes<GetObjectResponse> objectBytes =
 s3.getObjectAsBytes(objectRequest);
            byte[] data = objectBytes.asByteArray();

            // Write the data to a local file.
            File myFile = new File(path);
            OutputStream os = new FileOutputStream(myFile);
            os.write(data);
            System.out.println("Successfully obtained bytes from an S3 object");
            os.close();

        } catch (IOException ex) {
            ex.printStackTrace();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Utilisez un S3 TransferManager pour télécharger un objet d'un compartiment S3 vers un fichier local. Consultez le fichier complet et le test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadFileRequest;
import software.amazon.awssdk.transfer.s3.model.FileDownload;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;

import java.io.IOException;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
```

```
import java.util.UUID;

    public Long downloadFile(S3TransferManager transferManager, String
 bucketName,
                             String key, String downloadedFileWithPath) {
        DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
                .getObjectRequest(b -> b.bucket(bucketName).key(key))
                .destination(Paths.get(downloadedFileWithPath))
                .build();

        FileDownload downloadFile =
 transferManager.downloadFile(downloadFileRequest);

        CompletedFileDownload downloadResult =
 downloadFile.completionFuture().join();
        logger.info("Content length [{}]",
 downloadResult.response().contentLength());
        return downloadResult.response().contentLength();
    }
```

Lisez les étiquettes qui appartiennent à un objet à l'aide d'un S3Client.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Tag;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetObjectTags {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:
                    <bucketName> <keyName>\s

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - A key name that represents the object.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        listTags(s3, bucketName, keyName);
        s3.close();
    }

    public static void listTags(S3Client s3, String bucketName, String keyName) {
        try {
            GetObjectTaggingRequest getTaggingRequest = GetObjectTaggingRequest
                    .builder()
                    .key(keyName)
                    .bucket(bucketName)
                    .build();

            GetObjectTaggingResponse tags =
s3.getObjectTagging(getTaggingRequest);
            List<Tag> tagSet = tags.tagSet();
            for (Tag tag : tagSet) {
                System.out.println(tag.key());
                System.out.println(tag.value());
            }

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
```

```
            System.exit(1);
        }
    }
}
```

Obtenez une URL pour un objet en utilisant un S3Client.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetUrlRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.net.URL;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetObjectUrl {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName>\s

                Where:
                    bucketName - The Amazon S3 bucket name.
                    keyName - A key name that represents the object.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
```

```
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getURL(s3, bucketName, keyName);
        s3.close();
    }

    public static void getURL(S3Client s3, String bucketName, String keyName) {
        try {
            GetUrlRequest request = GetUrlRequest.builder()
                    .bucket(bucketName)
                    .key(keyName)
                    .build();

            URL url = s3.utilities().getUrl(request);
            System.out.println("The URL for  " + keyName + " is " + url);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Obtenez un objet en utilisant l'objet client S3Presigner via un S3Client.

```
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.time.Duration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import
 software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
 software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import software.amazon.awssdk.utils.IoUtils;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetObjectPresignedUrl {
    public static void main(String[] args) {
        final String USAGE = """

                Usage:
                    <bucketName> <keyName>\s

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - A key name that represents a text file.\s
                """;

        if (args.length != 2) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Presigner presigner = S3Presigner.builder()
                .region(region)
                .build();

        getPresignedUrl(presigner, bucketName, keyName);
        presigner.close();
    }

    public static void getPresignedUrl(S3Presigner presigner, String bucketName,
 String keyName) {
        try {
            GetObjectRequest getObjectRequest = GetObjectRequest.builder()
                    .bucket(bucketName)
                    .key(keyName)
```

```
                    .build();

            GetObjectPresignRequest getObjectPresignRequest =
GetObjectPresignRequest.builder()
                    .signatureDuration(Duration.ofMinutes(60))
                    .getObjectRequest(getObjectRequest)
                    .build();

            PresignedGetObjectRequest presignedGetObjectRequest =
presigner.presignGetObject(getObjectPresignRequest);
            String theUrl = presignedGetObjectRequest.url().toString();
            System.out.println("Presigned URL: " + theUrl);
            HttpURLConnection connection = (HttpURLConnection)
presignedGetObjectRequest.url().openConnection();
            presignedGetObjectRequest.httpRequest().headers().forEach((header,
values) -> {
                values.forEach(value -> {
                    connection.addRequestProperty(header, value);
                });
            });

            // Send any request payload that the service needs (not needed when
            // isBrowserExecutable is true).
            if (presignedGetObjectRequest.signedPayload().isPresent()) {
                connection.setDoOutput(true);

                try (InputStream signedPayload =
presignedGetObjectRequest.signedPayload().get().asInputStream();
                        OutputStream httpOutputStream =
connection.getOutputStream()) {
                    IoUtils.copy(signedPayload, httpOutputStream);
                }
            }

            // Download the result of executing the request.
            try (InputStream content = connection.getInputStream()) {
                System.out.println("Service returned response: ");
                IoUtils.copy(content, System.out);
            }

        } catch (S3Exception | IOException e) {
            e.getStackTrace();
        }
    }
```

```
}
```

Obtenez un objet en utilisant un ResponseTransformer objet et S3Client.

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.core.sync.ResponseTransformer;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class GetDataResponseTransformer {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName> <path>

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - The key name.\s
                    path - The path where the file is written to.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
```

```
        }

        String bucketName = args[0];
        String keyName = args[1];
        String path = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getObjectBytes(s3, bucketName, keyName, path);
        s3.close();
    }

    public static void getObjectBytes(S3Client s3, String bucketName, String
 keyName, String path) {
        try {
            GetObjectRequest objectRequest = GetObjectRequest
                    .builder()
                    .key(keyName)
                    .bucket(bucketName)
                    .build();

            ResponseBytes<GetObjectResponse> objectBytes =
 s3.getObject(objectRequest, ResponseTransformer.toBytes());
            byte[] data = objectBytes.asByteArray();

            // Write the data to a local file.
            File myFile = new File(path);
            OutputStream os = new FileOutputStream(myFile);
            os.write(data);
            System.out.println("Successfully obtained bytes from an S3 object");
            os.close();

        } catch (IOException ex) {
            ex.printStackTrace();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Téléchargez l'objet.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new GetObjectCommand({
    Bucket: "test-bucket",
    Key: "hello-s3.txt",
  });

  try {
    const response = await client.send(command);
    // The Body object also has 'transformToByteArray' and 'transformToWebStream'
 methods.
    const str = await response.Body.transformToString();
    console.log(str);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun getObjectBytes(
    bucketName: String,
    keyName: String,
    path: String,
) {
    val request =
        GetObjectRequest {
            key = keyName
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.getObject(request) { resp ->
            val myFile = File(path)
            resp.body?.writeToFile(myFile)
            println("Successfully read $keyName from $bucketName")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez un objet.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $file = $this->s3client->getObject([
                'Bucket' => $this->bucketName,
                'Key' => $fileName,
            ]);
            $body = $file->get('Body');
            $body->rewind();
            echo "Downloaded the file and it begins with: {$body->read(26)}.\n";
        } catch (Exception $exception) {
            echo "Failed to download $fileName from $this->bucketName with error:
" . $exception->getMessage();
            exit("Please fix error with file downloading before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for PHP API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande récupère l'élément « sample.txt » du bucket « test-files » et l'enregistre dans un fichier nommé « local-sample.txt » à l'emplacement actuel. Il n'est pas nécessaire que le fichier « local-sample.txt » existe pour que cette commande soit appelée.

```
Read-S3Object -BucketName test-files -Key sample.txt -File local-sample.txt
```

Exemple 2 : Cette commande extrait le répertoire virtuel « DIR » du bucket « test-files » et l'enregistre dans un dossier nommé « Local-dir » à l'emplacement actuel. Le dossier « Local-dir » n'a pas besoin d'exister pour que cette commande soit appelée.

```
Read-S3Object -BucketName test-files -KeyPrefix DIR -Folder Local-DIR
```

Exemple 3 : télécharge tous les objets dont les clés se terminent par « .json » depuis les compartiments dont le nom contient « config » vers les fichiers du dossier spécifié. Les clés d'objet sont utilisées pour définir les noms de fichiers.

```
Get-S3Bucket | ? { $_.BucketName -like '*config*' } | Get-S3Object | ? { $_.Key -
like '*.json' } | Read-S3Object -Folder C:\ConfigObjects
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key
```

```
    def get(self):
        """
        Gets the object.

        :return: The object data in bytes.
        """
        try:
            body = self.object.get()["Body"].read()
            logger.info(
                "Got object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't get object '%s' from bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
            raise
        else:
            return body
```

- Pour plus de détails sur l'API, consultez GetObjectle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez un objet.

```
require "aws-sdk-s3"
```

```
# Wraps Amazon S3 object actions.
class ObjectGetWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object directly to a file.
  #
  # @param target_path [String] The path to the file where the object is
 downloaded.
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def get_object(target_path)
    @object.get(response_target: target_path)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"
  target_path = "my-object-as-file.txt"

  wrapper = ObjectGetWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  obj_data = wrapper.get_object(target_path)
  return unless obj_data

  puts "Object #{object_key} (#{obj_data.content_length} bytes} downloaded to
 #{target_path}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Obtenez un objet et signalez son état de chiffrement côté serveur.

```
require "aws-sdk-s3"
```

```
# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def get_object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? "no" :
 obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn get_object(client: Client, opt: Opt) -> Result<usize, anyhow::Error> {
    trace!("bucket:      {}", opt.bucket);
    trace!("object:      {}", opt.object);
    trace!("destination: {}", opt.destination.display());

    let mut file = File::create(opt.destination.clone())?;

    let mut object = client
        .get_object()
        .bucket(opt.bucket)
        .key(opt.object)
        .send()
        .await?;

    let mut byte_count = 0_usize;
    while let Some(bytes) = object.body.try_next().await? {
        let bytes_len = bytes.len();
        file.write_all(&bytes)?;
        trace!("Intermediate write of {bytes_len}");
        byte_count += bytes_len;
    }

    Ok(byte_count)
}
```

- Pour plus de détails sur l'API, voir GetObjectla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   TRY.
       oo_result = lo_s3->getobject(           " oo_result is returned for
testing purposes. "
                   iv_bucket = iv_bucket_name
                   iv_key = iv_object_key
                 ).
       DATA(lv_object_data) = oo_result->get_body( ).
       MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
     CATCH /aws1/cx_s3_nosuchkey.
       MESSAGE 'Object key does not exist.' TYPE 'E'.
   ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d’être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Téléchargez un objet d'un compartiment vers un fichier local.

```
    public func downloadFile(bucket: String, key: String, to: String) async
throws {
        let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the data stream object. Return immediately if there isn't one.
        guard let body = output.body,
              let data = try await body.readData() else {
            return
        }
        try data.write(to: fileUrl)
    }
```

Lisez un objet dans un objet Swift Data.

```
    public func readFile(bucket: String, key: String) async throws -> Data {
        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the stream and return its contents in a `Data` object. If
        // there is no stream, return an empty `Data` object instead.
        guard let body = output.body,
              let data = try await body.readData() else {
            return "".data(using: .utf8)!
        }
```

```
        return data
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObjectAcl`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObjectAcl`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Gérer les listes de contrôle d'accès (ACL)

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::getObjectAcl(const Aws::String &bucketName,
                              const Aws::String &objectKey,
                              const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::GetObjectAclRequest request;
    request.SetBucket(bucketName);
```

```
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectAclOutcome outcome =
            s3Client.GetObjectAcl(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: getObjectAcl: "
                  << err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
    } else {
        Aws::Vector<Aws::S3::Model::Grant> grants =
                outcome.GetResult().GetGrants();

        for (auto it = grants.begin(); it != grants.end(); it++) {
            std::cout << "For object " << objectKey << ": "
                      << std::endl << std::endl;

            Aws::S3::Model::Grant grant = *it;
            Aws::S3::Model::Grantee grantee = grant.GetGrantee();

            if (grantee.TypeHasBeenSet()) {
                std::cout << "Type:          "
                          << getGranteeTypeString(grantee.GetType()) <<
std::endl;
            }

            if (grantee.DisplayNameHasBeenSet()) {
                std::cout << "Display name:  "
                          << grantee.GetDisplayName() << std::endl;
            }

            if (grantee.EmailAddressHasBeenSet()) {
                std::cout << "Email address: "
                          << grantee.GetEmailAddress() << std::endl;
            }

            if (grantee.IDHasBeenSet()) {
                std::cout << "ID:            "
                          << grantee.GetID() << std::endl;
            }

            if (grantee.URIHasBeenSet()) {
                std::cout << "URI:           "
```

```
                            << grantee.GetURI() << std::endl;
            }

            std::cout << "Permission:    " <<
                      getPermissionString(grant.GetPermission()) <<
                      std::endl << std::endl;
        }
    }

    return outcome.IsSuccess();
}

//! Routine which converts a built-in type enumeration to a human-readable
 string.
/*!
 \param type: Type enumeration.
 \return String: Human-readable string
*/
Aws::String getGranteeTypeString(const Aws::S3::Model::Type &type) {
    switch (type) {
        case Aws::S3::Model::Type::AmazonCustomerByEmail:
            return "Email address of an AWS account";
        case Aws::S3::Model::Type::CanonicalUser:
            return "Canonical user ID of an AWS account";
        case Aws::S3::Model::Type::Group:
            return "Predefined Amazon S3 group";
        case Aws::S3::Model::Type::NOT_SET:
            return "Not set";
        default:
            return "Type unknown";
    }
}

//! Routine which converts a built-in type enumeration to a human-readable
 string.
/*!
 \param permission: Permission enumeration.
 \return String: Human-readable string
*/
Aws::String getPermissionString(const Aws::S3::Model::Permission &permission) {
    switch (permission) {
        case Aws::S3::Model::Permission::FULL_CONTROL:
            return "Can read this object's data and its metadata, "
                   "and read/write this object's permissions";
```

```
        case Aws::S3::Model::Permission::NOT_SET:
            return "Permission not set";
        case Aws::S3::Model::Permission::READ:
            return "Can read this object's data and its metadata";
        case Aws::S3::Model::Permission::READ_ACP:
            return "Can read this object's permissions";
            // case Aws::S3::Model::Permission::WRITE // Not applicable.
        case Aws::S3::Model::Permission::WRITE_ACP:
            return "Can write this object's permissions";
        default:
            return "Permission unknown";
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectAclà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante permet de récupérer la liste de contrôle d'accès pour un objet dans un compartiment nommé `my-bucket` :

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

Sortie :

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
```

```
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectAclà la section Référence des AWS CLI commandes.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun getBucketACL(
    objectKey: String,
    bucketName: String,
) {
    val request =
        GetObjectAclRequest {
            bucket = bucketName
            key = objectKey
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.getObjectAcl(request)
        response.grants?.forEach { grant ->
            println("Grant permission is ${grant.permission}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectAclà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    def get_acl(self):
        """
        Gets the ACL of the object.

        :return: The ACL of the object.
        """
        try:
            acl = self.object.Acl()
            logger.info(
                "Got ACL for object %s owned by %s.",
                self.object.key,
                acl.owner["DisplayName"],
            )
```

```
        except ClientError:
            logger.exception("Couldn't get ACL for object %s.", self.object.key)
            raise
        else:
            return acl
```

- Pour plus de détails sur l'API, consultez GetObjectAcllle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObjectLegalHold`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObjectLegalHold`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Obtenir la configuration légale de conservation d'un objet
- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    /// <summary>
    /// Get the legal hold details for an S3 object.
```

```
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The object key.</param>
    /// <returns>The object legal hold details.</returns>
    public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
        string objectKey)
    {
        try
        {
            var request = new GetObjectLegalHoldRequest()
            {
                BucketName = bucketName,
                Key = objectKey
            };

            var response = await _amazonS3.GetObjectLegalHoldAsync(request);
            Console.WriteLine($"\tObject legal hold for {objectKey} in
{bucketName}: " +
                              $"\n\tStatus: {response.LegalHold.Status}");
            return response.LegalHold;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch legal hold: '{ex.Message}'");
            return new ObjectLockLegalHold();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Récupère le statut de conservation légale d'un objet

L'`get-object-legal-hold`exemple suivant permet de récupérer le statut Legal Hold pour l'objet spécifié.

```
aws s3api get-object-legal-hold \
    --bucket my-bucket-with-object-lock \
```

```
    --key doc1.rtf
```

Sortie :

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// GetObjectLegalHold retrieves the legal hold status for an S3 object.
func (actor S3Actions) GetObjectLegalHold(ctx context.Context, bucket string, key
 string, versionId string) (*types.ObjectLockLegalHoldStatus, error) {
 var status *types.ObjectLockLegalHoldStatus
 input := &s3.GetObjectLegalHoldInput{
  Bucket:    aws.String(bucket),
  Key:       aws.String(key),
  VersionId: aws.String(versionId),
 }
```

```
 output, err := actor.S3Client.GetObjectLegalHold(ctx, input)
 if err != nil {
  var noSuchKeyErr *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noSuchKeyErr) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noSuchKeyErr
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
   case "NoSuchObjectLockConfiguration":
    log.Printf("Object %s does not have an object lock configuration.\n", key)
    err = nil
   case "InvalidRequest":
    log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
    err = nil
   }
  }
 } else {
  status = &output.LegalHold.Status
 }

 return status, err
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    // Get the legal hold details for an S3 object.
```

```
   public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
       try {
           GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
               .bucket(bucketName)
               .key(objectKey)
               .build();

           GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
           System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
               ":\n\tStatus: " + response.legalHold().status());
           return response.legalHold();

       } catch (S3Exception ex) {
           System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
       }

       return null;
   }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObjectLockConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObjectLockConfiguration`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   /// <summary>
   /// Get the object lock configuration details for an S3 bucket.
   /// </summary>
   /// <param name="bucketName">The bucket to get details.</param>
   /// <returns>The bucket's object lock configuration details.</returns>
   public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
   {
       try
       {
           var request = new GetObjectLockConfigurationRequest()
           {
               BucketName = bucketName
           };

           var response = await
_amazonS3.GetObjectLockConfigurationAsync(request);
           Console.WriteLine($"\tBucket object lock config for {bucketName} in
{bucketName}: " +
                             $"\n\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
                             $"\n\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

           return response.ObjectLockConfiguration;
       }
       catch (AmazonS3Exception ex)
       {
           Console.WriteLine($"\tUnable to fetch object lock config:
'{ex.Message}'");
```

```
            return new ObjectLockConfiguration();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour récupérer une configuration de verrouillage d'objet pour un bucket

L'`get-object-lock-configuration`exemple suivant récupère la configuration du verrouillage d'objet pour le compartiment spécifié.

```
aws s3api get-object-lock-configuration \
    --bucket my-bucket-with-object-lock
```

Sortie :

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// GetObjectLockConfiguration retrieves the object lock configuration for an S3
 bucket.
func (actor S3Actions) GetObjectLockConfiguration(ctx context.Context, bucket
 string) (*types.ObjectLockConfiguration, error) {
 var lockConfig *types.ObjectLockConfiguration
 input := &s3.GetObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
 }

 output, err := actor.S3Client.GetObjectLockConfiguration(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  } else if errors.As(err, &apiErr) && apiErr.ErrorCode() ==
 "ObjectLockConfigurationNotFoundError" {
   log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
   err = nil
  }
 } else {
  lockConfig = output.ObjectLockConfiguration
```

```
 }

 return lockConfig, err
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   // Get the object lock configuration details for an S3 bucket.
   public void getBucketObjectLockConfiguration(String bucketName) {
       GetObjectLockConfigurationRequest objectLockConfigurationRequest =
GetObjectLockConfigurationRequest.builder()
           .bucket(bucketName)
           .build();

       GetObjectLockConfigurationResponse response =
getClient().getObjectLockConfiguration(objectLockConfigurationRequest);
       System.out.println("Bucket object lock config for "+bucketName +":  ");
       System.out.println("\tEnabled:
"+response.objectLockConfiguration().objectLockEnabled());
       System.out.println("\tRule: "+
response.objectLockConfiguration().rule().defaultRetention());
   }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import {
  GetObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new GetObjectLockConfigurationCommand({
    Bucket: bucketName,
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
  });

  try {
    const { ObjectLockConfiguration } = await client.send(command);
    console.log(`Object Lock Configuration: ${ObjectLockConfiguration}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie la valeur « Enabled » si la configuration Object Lock est activée pour le compartiment S3 donné.

```
Get-S3ObjectLockConfiguration -BucketName 's3buckettesting' -Select
 ObjectLockConfiguration.ObjectLockEnabled
```

Sortie :

```
Value
-----
Enabled
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLockConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObjectRetention`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObjectRetention`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    /// <summary>
    /// Get the retention period for an S3 object.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The object key.</param>
    /// <returns>The object retention details.</returns>
    public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
        string objectKey)
    {
        try
        {
            var request = new GetObjectRetentionRequest()
            {
                BucketName = bucketName,
                Key = objectKey
            };

            var response = await _amazonS3.GetObjectRetentionAsync(request);
            Console.WriteLine($"\tObject retention for {objectKey} in
{bucketName}: " +
                              $"\n\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}.");
            return response.Retention;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch object lock retention:
'{ex.Message}'");
            return new ObjectLockRetention();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour récupérer la configuration de rétention d'un objet

L'`get-object-retention`exemple suivant récupère la configuration de rétention d'objets pour l'objet spécifié.

```
aws s3api get-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

Sortie :

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
```

```
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// GetObjectRetention retrieves the object retention configuration for an S3
 object.
func (actor S3Actions) GetObjectRetention(ctx context.Context, bucket string, key
 string) (*types.ObjectLockRetention, error) {
 var retention *types.ObjectLockRetention
 input := &s3.GetObjectRetentionInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
 }

 output, err := actor.S3Client.GetObjectRetention(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
   case "NoSuchObjectLockConfiguration":
    err = nil
   case "InvalidRequest":
    log.Printf("Bucket %s does not have locking enabled.", bucket)
    err = nil
   }
  }
 } else {
  retention = output.Retention
 }

 return retention, err
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   // Get the retention period for an S3 object.
   public ObjectLockRetention getObjectRetention(String bucketName, String key){
       try {
           GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();

           GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
           System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
           return response.retention();

       } catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           return null;
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import { GetObjectRetentionCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new GetObjectRetentionCommand({
    Bucket: bucketName,
    Key: objectKey,
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const { Retention } = await client.send(command);
    console.log(`Object Retention Settings: ${Retention.Status}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande renvoie le mode et la date jusqu'à ce que l'objet soit conservé.

```
Get-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt'
```

- Pour plus de détails sur l'API, reportez-vous GetObjectRetentionà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetObjectTagging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetObjectTagging`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Démarrer avec les étiquettes

CLI

AWS CLI

Pour récupérer les tags attachés à un objet

L'`get-object-tagging`exemple suivant extrait les valeurs de la clé spécifiée à partir de l'objet spécifié.

```
aws s3api get-object-tagging \
```

```
    --bucket my-bucket \
    --key doc1.rtf
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        }
    ]
}
```

L'`get-object-tagging`exemple suivant essaie de récupérer les ensembles de balises de l'objet`doc2.rtf`, qui ne possède aucune balise.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc2.rtf
```

Sortie :

```
{
    "TagSet": []
}
```

L'`get-object-tagging`exemple suivant récupère les ensembles de balises de l'objet`doc3.rtf`, qui possède plusieurs balises.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc3.rtf
```

Sortie :

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
```

```
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectTaggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : L'exemple renvoie les balises associées à l'objet présent dans le compartiment S3 donné.

```
Get-S3ObjectTagSet -Key 'testfile.txt' -BucketName 'testbucket123'
```

Sortie :

```
Key  Value
---  -----
test value
```

- Pour plus de détails sur l'API, reportez-vous GetObjectTaggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetPublicAccessBlock`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`GetPublicAccessBlock`.

CLI

AWS CLI

Pour définir ou modifier la configuration de blocage de l'accès public pour un bucket

L'`get-public-access-block`exemple suivant montre la configuration de blocage de l'accès public pour le compartiment spécifié.

```
aws s3api get-public-access-block \
    --bucket my-bucket
```

Sortie :

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetPublicAccessBlockà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande renvoie la configuration du bloc d'accès public du compartiment S3 donné.

```
Get-S3PublicAccessBlock -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous GetPublicAccessBlockà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **HeadBucket** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliserHeadBucket.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#############################################################################
# function bucket_exists
#
# This function checks to see if the specified bucket already exists.
#
# Parameters:
#       $1 - The name of the bucket to check.
#
# Returns:
#       0 - If the bucket already exists.
#       1 - If the bucket doesn't exist.
#############################################################################
function bucket_exists() {
  local bucket_name
  bucket_name=$1

  # Check whether the bucket already exists.
  # We suppress all output - we're interested only in the return code.

  if aws s3api head-bucket \
    --bucket "$bucket_name" \
    >/dev/null 2>&1; then
    return 0 # 0 in Bash script means true.
  else
```

```
    return 1 # 1 in Bash script means false.
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous HeadBucketà la section Référence des AWS CLI commandes.

CLI

AWS CLI

La commande suivante vérifie l'accès à un compartiment nommé `my-bucket` :

```
aws s3api head-bucket --bucket my-bucket
```

Si le bucket existe et que vous y avez accès, aucune sortie n'est renvoyée. Dans le cas contraire, un message d'erreur s'affichera. Par exemple :

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- Pour plus de détails sur l'API, reportez-vous HeadBucketà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
```

```
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
 _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
  Bucket: aws.String(bucketName),
 })
 exists := true
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NotFound:
    log.Printf("Bucket %v is available.\n", bucketName)
    exists = false
    err = nil
   default:
    log.Printf("Either you don't have access to bucket %v or another error
 occurred. "+
     "Here's what happened: %v\n", bucketName, err)
   }
  }
 } else {
  log.Printf("Bucket %v exists and you already own it.", bucketName)
 }

 return exists, err
}
```

- Pour plus de détails sur l'API, reportez-vous HeadBucketà la section Référence des AWS SDK for Go API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def exists(self):
        """
        Determine whether the bucket exists and you have access to it.

        :return: True when the bucket exists; otherwise, False.
        """
        try:
            self.bucket.meta.client.head_bucket(Bucket=self.bucket.name)
            logger.info("Bucket %s exists.", self.bucket.name)
            exists = True
        except ClientError:
            logger.warning(
                "Bucket %s doesn't exist or you don't have access to it.",
                self.bucket.name,
            )
            exists = False
        return exists
```

- Pour plus de détails sur l'API, consultez HeadBucketle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`HeadObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`HeadObject`.

CLI

AWS CLI

La commande suivante permet de récupérer les métadonnées d'un objet dans un compartiment nommé `my-bucket` :

```
aws s3api head-object --bucket my-bucket --key index.html
```

Sortie :

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- Pour plus de détails sur l'API, reportez-vous HeadObjectà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Déterminez le type de contenu d'un objet.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class GetObjectContentType {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName>>

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - The key name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        getContentType(s3, bucketName, keyName);
        s3.close();
    }

    public static void getContentType(S3Client s3, String bucketName, String
 keyName) {
        try {
            HeadObjectRequest objectRequest = HeadObjectRequest.builder()
                    .key(keyName)
                    .bucket(bucketName)
                    .build();

            HeadObjectResponse objectHead = s3.headObject(objectRequest);
            String type = objectHead.contentType();
            System.out.println("The object content type is " + type);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

Obtenez le statut de restauration d'un objet.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;

public class GetObjectRestoreStatus {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                    <bucketName> <keyName>\s

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - A key name that represents the object.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        checkStatus(s3, bucketName, keyName);
        s3.close();
    }

    public static void checkStatus(S3Client s3, String bucketName, String
 keyName) {
        try {
            HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
                    .bucket(bucketName)
                    .key(keyName)
                    .build();

            HeadObjectResponse response = s3.headObject(headObjectRequest);
            System.out.println("The Amazon S3 object restoration status is " +
 response.restore());

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous HeadObjectà la section Référence des AWS SDK for Java 2.x API.

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectExistsWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Checks whether the object exists.
  #
  # @return [Boolean] True if the object exists; otherwise false.
  def exists?
    @object.exists?
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't check existence of object
 #{@object.bucket.name}:#{@object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
```

```
  object_key = "my-object.txt"

  wrapper = ObjectExistsWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  exists = wrapper.exists?

  puts "Object #{object_key} #{exists ? 'does' : 'does not'} exist."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous HeadObjectà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListBucketAnalyticsConfigurations`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListBucketAnalyticsConfigurations`.

CLI

AWS CLI

Pour récupérer la liste des configurations d'analyse pour un bucket

Ce qui suit `list-bucket-analytics-configurations` permet de récupérer la liste des configurations d'analyse pour le compartiment spécifié.

```
aws s3api list-bucket-analytics-configurations \
    --bucket my-bucket
```

Sortie :

```
{
```

```
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketAnalyticsConfigurationsà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les 100 premières configurations d'analyse du compartiment S3 donné.

```
Get-S3BucketAnalyticsConfigurationList -BucketName 's3casetestbucket'
```

- Pour plus de détails sur l'API, reportez-vous ListBucketAnalyticsConfigurationsà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListBucketInventoryConfigurations`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListBucketInventoryConfigurations`.

CLI

AWS CLI

Pour récupérer la liste des configurations d'inventaire pour un bucket

L'`list-bucket-inventory-configurations`exemple suivant répertorie les configurations d'inventaire pour le compartiment spécifié.

```
aws s3api list-bucket-inventory-configurations \
    --bucket my-bucket
```

Sortie :

```
{
    "InventoryConfigurationList": [
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "ORC",
                    "Bucket": "arn:aws:s3:::my-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "1",
            "Schedule": {
                "Frequency": "Weekly"
            }
        },
        {
            "IsEnabled": true,
            "Destination": {
                "S3BucketDestination": {
                    "Format": "CSV",
                    "Bucket": "arn:aws:s3:::my-bucket",
                    "AccountId": "123456789012"
                }
            },
            "IncludedObjectVersions": "Current",
            "Id": "2",
            "Schedule": {
                "Frequency": "Daily"
            }
        }
    ],
    "IsTruncated": false
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketInventoryConfigurationsà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie les 100 premières configurations d'inventaire du compartiment S3 donné.

```
Get-S3BucketInventoryConfigurationList -BucketName 's3testbucket'
```

- Pour plus de détails sur l'API, reportez-vous ListBucketInventoryConfigurationsà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListBuckets`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListBuckets`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
namespace ListBucketsExample
{
    using System;
    using System.Collections.Generic;
```

```
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example uses the AWS SDK for .NET to list the Amazon Simple Storage
    /// Service (Amazon S3) buckets belonging to the default account.
    /// </summary>
    public class ListBuckets
    {
        private static IAmazonS3 _s3Client;

        /// <summary>
        /// Get a list of the buckets owned by the default user.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <returns>The response from the ListingBuckets call that contains a
        /// list of the buckets owned by the default user.</returns>
        public static async Task<ListBucketsResponse> GetBuckets(IAmazonS3
client)
        {
            return await client.ListBucketsAsync();
        }

        /// <summary>
        /// This method lists the name and creation date for the buckets in
        /// the passed List of S3 buckets.
        /// </summary>
        /// <param name="bucketList">A List of S3 bucket objects.</param>
        public static void DisplayBucketList(List<S3Bucket> bucketList)
        {
            bucketList
                .ForEach(b => Console.WriteLine($"Bucket name: {b.BucketName},
created on: {b.CreationDate}"));
        }

        public static async Task Main()
        {
            // The client uses the AWS Region of the default user.
            // If the Region where the buckets were created is different,
            // pass the Region to the client constructor. For example:
            // _s3Client = new AmazonS3Client(RegionEndpoint.USEast1);
            _s3Client = new AmazonS3Client();
            var response = await GetBuckets(_s3Client);
```

```
            DisplayBucketList(response.Buckets);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::listBuckets(const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client client(clientConfig);

    auto outcome = client.ListBuckets();

    bool result = true;
    if (!outcome.IsSuccess()) {
        std::cerr << "Failed with error: " << outcome.GetError() << std::endl;
        result = false;
    } else {
        std::cout << "Found " << outcome.GetResult().GetBuckets().size() << "
 buckets\n";
        for (auto &&b: outcome.GetResult().GetBuckets()) {
            std::cout << b.GetName() << std::endl;
        }
    }

    return result;
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante utilise la `list-buckets` commande pour afficher les noms de tous vos compartiments Amazon S3 (dans toutes les régions) :

```
aws s3api list-buckets --query "Buckets[].Name"
```

L'option de requête filtre la sortie en ne `list-buckets` reportant que les noms des compartiments.

Pour plus d'informations sur les compartiments, consultez la section Utilisation des compartiments Amazon S3 dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
```

```
}



// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
 result, err := basics.S3Client.ListBuckets(context.TODO(),
 &s3.ListBucketsInput{})
 var buckets []types.Bucket
 if err != nil {
  log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
 } else {
  buckets = result.Buckets
 }
 return buckets, err
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
```

```
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListBuckets {
    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
            .region(region)
            .build();

        listAllBuckets(s3);

    }
    public static void listAllBuckets(S3Client s3) {
        ListBucketsResponse response = s3.listBuckets();
        List<Bucket> bucketList = response.buckets();
        for (Bucket bucket: bucketList) {
            System.out.println("Bucket name "+bucket.name());
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Listez les compartiments.

```
import { ListBucketsCommand, S3Client } from "@aws-sdk/client-s3";
```

```
const client = new S3Client({});

export const main = async () => {
  const command = new ListBucketsCommand({});

  try {
    const { Owner, Buckets } = await client.send(command);
    console.log(
      `${Owner.DisplayName} owns ${Buckets.length} bucket${
        Buckets.length === 1 ? "" : "s"
      }:`,
    );
    console.log(`${Buckets.map((b) => ` • ${b.Name}`).join("\n")}`);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande renvoie tous les compartiments S3.

```
Get-S3Bucket
```

Exemple 2 : Cette commande renvoie un bucket nommé « test-files »

```
Get-S3Bucket -BucketName test-files
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    @staticmethod
    def list(s3_resource):
        """
        Get the buckets in all Regions for the current account.

        :param s3_resource: A Boto3 S3 resource. This is a high-level resource in
 Boto3
                            that contains collections and factory methods to
 create
                            other high-level S3 sub-resources.
        :return: The list of buckets.
        """
        try:
            buckets = list(s3_resource.buckets.all())
            logger.info("Got buckets: %s.", buckets)
        except ClientError:
            logger.exception("Couldn't get buckets.")
            raise
        else:
```

```
            return buckets
```

- Pour plus de détails sur l'API, consultez ListBucketsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 resource actions.
class BucketListWrapper
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
    @s3_resource = s3_resource
  end

  # Lists buckets for the current account.
  #
  # @param count [Integer] The maximum number of buckets to list.
  def list_buckets(count)
    puts "Found these buckets:"
    @s3_resource.buckets.each do |bucket|
      puts "\t#{bucket.name}"
      count -= 1
      break if count.zero?
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list buckets. Here's why: #{e.message}"
```

```
    false
  end
end

# Example usage:
def run_demo
  wrapper = BucketListWrapper.new(Aws::S3::Resource.new)
  wrapper.list_buckets(25)
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn show_buckets(strict: bool, client: &Client, region: &str) -> Result<(),
 Error> {
    let resp = client.list_buckets().send().await?;
    let buckets = resp.buckets();
    let num_buckets = buckets.len();

    let mut in_region = 0;

    for bucket in buckets {
        if strict {
            let r = client
                .get_bucket_location()
                .bucket(bucket.name().unwrap_or_default())
                .send()
                .await?;
```

```
            if r.location_constraint().unwrap().as_ref() == region {
                println!("{}", bucket.name().unwrap_or_default());
                in_region += 1;
            }
        } else {
            println!("{}", bucket.name().unwrap_or_default());
        }
    }

    println!();
    if strict {
        println!(
            "Found {} buckets in the {} region out of a total of {} buckets.",
            in_region, region, num_buckets
        );
    } else {
        println!("Found {} buckets in all regions.", num_buckets);
    }

    Ok(())
}
```

- Pour plus de détails sur l'API, voir ListBucketsla section de référence de l'API AWS SDK for Rust.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/// Return an array containing information about every available bucket.
///
/// - Returns: An array of ``S3ClientTypes.Bucket`` objects describing
///   each bucket.
public func getAllBuckets() async throws -> [S3ClientTypes.Bucket] {
    let output = try await client.listBuckets(input: ListBucketsInput())

    guard let buckets = output.buckets else {
        return []
    }
    return buckets
}
```

- Pour plus de détails sur l'API, reportez-vous ListBucketsà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListMultipartUploads`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListMultipartUploads`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Supprimer les téléchargements partitionnés incomplets

CLI

AWS CLI

La commande suivante répertorie tous les téléchargements partitionnés actifs pour un bucket nommé : `my-bucket`

```
aws s3api list-multipart-uploads --bucket my-bucket
```

Sortie :

```
{
    "Uploads": [
        {
            "Initiator": {
                "DisplayName": "username",
                "ID": "arn:aws:iam::0123456789012:user/username"
            },
            "Initiated": "2015-06-02T18:01:30.000Z",
            "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
            "StorageClass": "STANDARD",
            "Key": "multipart/01",
            "Owner": {
                "DisplayName": "aws-account-name",
                "ID":
 "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
            }
        }
    ],
    "CommonPrefixes": []
}
```

Les téléchargements partitionnés en cours entraînent des coûts de stockage dans Amazon S3. Terminez ou annulez un téléchargement en plusieurs parties actif pour en supprimer certaines parties de votre compte.

- Pour plus de détails sur l'API, reportez-vous ListMultipartUploadsà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsRequest;
import software.amazon.awssdk.services.s3.model.ListMultipartUploadsResponse;
import software.amazon.awssdk.services.s3.model.MultipartUpload;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ListMultipartUploads {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The name of the Amazon S3 bucket where an in-
progress multipart upload is occurring.
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();
        listUploads(s3, bucketName);
        s3.close();
    }

    public static void listUploads(S3Client s3, String bucketName) {
        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
 ListMultipartUploadsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListMultipartUploadsResponse response =
 s3.listMultipartUploads(listMultipartUploadsRequest);
            List<MultipartUpload> uploads = response.uploads();
            for (MultipartUpload upload : uploads) {
                System.out.println("Upload in progress: Key = \"" + upload.key()
 + "\", id = " + upload.uploadId());
            }

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListMultipartUploadsà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListObjectVersions`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListObjectVersions`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Utiliser les objets soumis au contrôle de version

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example lists the versions of the objects in a version enabled
/// Amazon Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class ListObjectVersions
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";

        // If the AWS Region where your bucket is defined is different from
        // the AWS Region where the Amazon S3 bucket is defined, pass the
constant
        // for the AWS Region to the client constructor like this:
        //      var client = new AmazonS3Client(RegionEndpoint.USWest2);
        IAmazonS3 client = new AmazonS3Client();
```

```
            await GetObjectListWithAllVersionsAsync(client, bucketName);
        }

        /// <summary>
        /// This method lists all versions of the objects within an Amazon S3
        /// version enabled bucket.
        /// </summary>
        /// <param name="client">The initialized client object used to call
        /// ListVersionsAsync.</param>
        /// <param name="bucketName">The name of the version enabled Amazon S3
 bucket
        /// for which you want to list the versions of the contained objects.</
param>
        public static async Task GetObjectListWithAllVersionsAsync(IAmazonS3
 client, string bucketName)
        {
            try
            {
                // When you instantiate the ListVersionRequest, you can
                // optionally specify a key name prefix in the request
                // if you want a list of object versions of a specific object.

                // For this example we set a small limit in MaxKeys to return
                // a small list of versions.
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    MaxKeys = 2,
                };

                do
                {
                    ListVersionsResponse response = await
 client.ListVersionsAsync(request);

                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine($"key: {entry.Key} size:
 {entry.Size}");
                    }

                    // If response is truncated, set the marker to get the next
                    // set of keys.
```

```
                    if (response.IsTruncated)
                    {
                        request.KeyMarker = response.NextKeyMarker;
                        request.VersionIdMarker = response.NextVersionIdMarker;
                    }
                    else
                    {
                        request = null;
                    }
                }
                while (request != null);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error: '{ex.Message}'");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous ListObjectVersionsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante permet de récupérer les informations de version d'un objet dans un compartiment nommé `my-bucket` :

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

Sortie :

```
{
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
```

```
            },
            "IsLatest": true,
            "VersionId": "B2VsEK5saUNNHKcOAJj7hIE86RozToyq",
            "Key": "index.html",
            "LastModified": "2015-11-10T00:57:03.000Z"
        },
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "VersionId": ".FLQEZscLIcfxSq.jsFJ.szUkmng2Yw6",
            "Key": "index.html",
            "LastModified": "2015-11-09T23:32:20.000Z"
        }
    ],
    "Versions": [
        {
            "LastModified": "2015-11-10T00:20:11.000Z",
            "VersionId": "Rb_l2T8UHDkFEwCgJjhlgPOZC0qJ.vpD",
            "ETag": "\"0622528de826c0df5db1258a23b80be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T23:26:41.000Z",
            "VersionId": "rasWWGpgk9E4s0LyTJgusGeRQKLVIAFf",
            "ETag": "\"06225825b8028de826c0df5db1a23be5\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
"7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
```

```
            "IsLatest": false,
            "Size": 38
        },
        {
            "LastModified": "2015-11-09T22:50:50.000Z",
            "VersionId": "null",
            "ETag": "\"d1f45267a863c8392e07d24dd592f1b9\"",
            "StorageClass": "STANDARD",
            "Key": "index.html",
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": false,
            "Size": 533823
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListObjectVersionsà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}
```

```
// ListObjectVersions lists all versions of all objects in a bucket.
func (actor S3Actions) ListObjectVersions(ctx context.Context, bucket string)
 ([]types.ObjectVersion, error) {
 var err error
 var output *s3.ListObjectVersionsOutput
 var versions []types.ObjectVersion
 input := &s3.ListObjectVersionsInput{Bucket: aws.String(bucket)}
 versionPaginator := s3.NewListObjectVersionsPaginator(actor.S3Client, input)
 for versionPaginator.HasMorePages() {
  output, err = versionPaginator.NextPage(ctx)
  if err != nil {
   var noBucket *types.NoSuchBucket
   if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
   }
   break
  } else {
   versions = append(versions, output.Versions...)
  }
 }
 return versions, err
}
```

- Pour plus de détails sur l'API, reportez-vous ListObjectVersionsà la section Référence des AWS SDK for Go API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
async fn show_versions(client: &Client, bucket: &str) -> Result<(), Error> {
```

```
    let resp = client.list_object_versions().bucket(bucket).send().await?;

    for version in resp.versions() {
        println!("{}", version.key().unwrap_or_default());
        println!("  version ID: {}", version.version_id().unwrap_or_default());
        println!();
    }

    Ok(())
}
```

- Pour plus de détails sur l'API, voir ListObjectVersionsla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListObjects`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListObjects`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Créer une page web qui répertorie les objets Amazon S3

CLI

AWS CLI

L'exemple suivant utilise la `list-objects` commande pour afficher les noms de tous les objets du compartiment spécifié :

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

L'exemple utilise l'`--query`argument pour filtrer la sortie `list-objects` jusqu'à la valeur clé et à la taille de chaque objet.

Pour plus d'informations sur les objets, consultez la section Travailler avec des objets Amazon S3 dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous ListObjectsà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande récupère les informations relatives à tous les éléments du bucket « test-files ».

```
Get-S3Object -BucketName test-files
```

Exemple 2 : Cette commande récupère les informations relatives à l'élément « sample.txt » depuis le bucket « test-files ».

```
Get-S3Object -BucketName test-files -Key sample.txt
```

Exemple 3 : Cette commande récupère les informations relatives à tous les éléments portant le préfixe « sample » à partir du bucket « test-files ».

```
Get-S3Object -BucketName test-files -KeyPrefix sample
```

- Pour plus de détails sur l'API, reportez-vous ListObjectsà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListObjectsV2`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`ListObjectsV2`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Démarrer avec les compartiments et les objets

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Shows how to list the objects in an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the bucket for which to list
        /// the contents.</param>
        /// <returns>A boolean value indicating the success or failure of the
        /// copy operation.</returns>
        public static async Task<bool> ListBucketContentsAsync(IAmazonS3 client,
string bucketName)
        {
            try
            {
                var request = new ListObjectsV2Request
                {
                    BucketName = bucketName,
                    MaxKeys = 5,
                };

                Console.WriteLine("--------------------------------------");
                Console.WriteLine($"Listing the contents of {bucketName}:");
                Console.WriteLine("--------------------------------------");

                ListObjectsV2Response response;
```

```
                do
                {
                    response = await client.ListObjectsV2Async(request);

                    response.S3Objects
                        .ForEach(obj => Console.WriteLine($"{obj.Key,-35}
{obj.LastModified.ToShortDateString(),10}{obj.Size,10}"));

                    // If the response is truncated, set the request
 ContinuationToken
                    // from the NextContinuationToken property of the response.
                    request.ContinuationToken = response.NextContinuationToken;
                }
                while (response.IsTruncated);

                return true;
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error encountered on server.
 Message:'{ex.Message}' getting list of objects.");
                return false;
            }
        }
```

Listez les objets avec un paginateur.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// The following example lists objects in an Amazon Simple Storage
    /// Service (Amazon S3) bucket.
    /// </summary>
    public class ListObjectsPaginator
    {
        private const string BucketName = "doc-example-bucket";
```

```
        public static async Task Main()
        {
            IAmazonS3 s3Client = new AmazonS3Client();

            Console.WriteLine($"Listing the objects contained in {BucketName}:
\n");
            await ListingObjectsAsync(s3Client, BucketName);
        }

        /// <summary>
        /// This method uses a paginator to retrieve the list of objects in an
        /// an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">An Amazon S3 client object.</param>
        /// <param name="bucketName">The name of the S3 bucket whose objects
        /// you want to list.</param>
        public static async Task ListingObjectsAsync(IAmazonS3 client, string
 bucketName)
        {
            var listObjectsV2Paginator = client.Paginators.ListObjectsV2(new
 ListObjectsV2Request
            {
                BucketName = bucketName,
            });

            await foreach (var response in listObjectsV2Paginator.Responses)
            {
                Console.WriteLine($"HttpStatusCode: {response.HttpStatusCode}");
                Console.WriteLine($"Number of Keys: {response.KeyCount}");
                foreach (var entry in response.S3Objects)
                {
                    Console.WriteLine($"Key = {entry.Key} Size = {entry.Size}");
                }
            }
        }
    }
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
```

```
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans AWS CLI Command Reference.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::listObjects(const Aws::String &bucketName,
                             const Aws::S3::S3ClientConfiguration &clientConfig)
 {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::ListObjectsV2Request request;
    request.WithBucket(bucketName);

    Aws::String continuationToken; // Used for pagination.
    Aws::Vector<Aws::S3::Model::Object> allObjects;

    do {
        if (!continuationToken.empty()) {
            request.SetContinuationToken(continuationToken);
        }

        auto outcome = s3Client.ListObjectsV2(request);
```

```
        if (!outcome.IsSuccess()) {
            std::cerr << "Error: listObjects: " <<
                      outcome.GetError().GetMessage() << std::endl;
            return false;
        } else {
            Aws::Vector<Aws::S3::Model::Object> objects =
                    outcome.GetResult().GetContents();

            allObjects.insert(allObjects.end(), objects.begin(), objects.end());
            continuationToken = outcome.GetResult().GetNextContinuationToken();
        }
    } while (!continuationToken.empty());

    std::cout << allObjects.size() << " object(s) found:" << std::endl;

    for (const auto &object: allObjects) {
        std::cout << "  " << object.GetKey() << std::endl;
    }

    return true;
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour obtenir la liste des objets d'un bucket

L'`list-objects-v2`exemple suivant répertorie les objets contenus dans le compartiment spécifié.

```
aws s3api list-objects-v2 \
    --bucket my-bucket
```

Sortie :

```
{
    "Contents": [
```

```
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"621503c373607d548b37cff8778d992c\"",
            "StorageClass": "STANDARD",
            "Key": "doc1.rtf",
            "Size": 391
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",
            "StorageClass": "STANDARD",
            "Key": "doc2.rtf",
            "Size": 373
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"08210852f65a2e9cb999972539a64d68\"",
            "StorageClass": "STANDARD",
            "Key": "doc3.rtf",
            "Size": 399
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"d1852dd683f404306569471af106988e\"",
            "StorageClass": "STANDARD",
            "Key": "doc4.rtf",
            "Size": 6225
        }
    ]
}
```


- Pour plus de détails sur l'API, voir ListObjectsV2 dans AWS CLI Command Reference.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
 {
 result, err := basics.S3Client.ListObjectsV2(context.TODO(),
 &s3.ListObjectsV2Input{
  Bucket: aws.String(bucketName),
 })
 var contents []types.Object
 if err != nil {
  log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
 err)
 } else {
  contents = result.Contents
 }
 return contents, err
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListObjectsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.S3Object;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ListObjects {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The Amazon S3 bucket from which objects are
 read.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        listBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void listBucketObjects(S3Client s3, String bucketName) {
        try {
            ListObjectsRequest listObjects = ListObjectsRequest
                    .builder()
                    .bucket(bucketName)
                    .build();

            ListObjectsResponse res = s3.listObjects(listObjects);
            List<S3Object> objects = res.contents();
            for (S3Object myValue : objects) {
                System.out.print("\n The name of the key is " + myValue.key());
                System.out.print("\n The object is " + calKb(myValue.size()) + "
 KBs");
                System.out.print("\n The owner is " + myValue.owner());
            }

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    // convert bytes to kbs.
    private static long calKb(Long val) {
        return val / 1024;
    }
}
```

Lister les objets en utilisant la pagination.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.paginators.ListObjectsV2Iterable;

public class ListObjectsPaginated {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The Amazon S3 bucket from which objects are
 read.\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        listBucketObjects(s3, bucketName);
        s3.close();
    }

    public static void listBucketObjects(S3Client s3, String bucketName) {
        try {
            ListObjectsV2Request listReq = ListObjectsV2Request.builder()
                    .bucket(bucketName)
                    .maxKeys(1)
                    .build();

            ListObjectsV2Iterable listRes = s3.listObjectsV2Paginator(listReq);
            listRes.stream()
                    .flatMap(r -> r.contents().stream())
```

```
                    .forEach(content -> System.out.println(" Key: " +
 content.key() + " size = " + content.size()));

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez tous les objets dans votre compartiment. S'il existe plusieurs objets, IsTruncated ils NextContinuationToken seront utilisés pour parcourir la liste complète.

```
import {
  S3Client,
  // This command supersedes the ListObjectsCommand and is the recommended way to
 list objects.
  ListObjectsV2Command,
} from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new ListObjectsV2Command({
    Bucket: "my-bucket",
    // The default and maximum number of keys returned is 1000. This limits it to
    // one for demonstration purposes.
```

```
    MaxKeys: 1,
  });

  try {
    let isTruncated = true;

    console.log("Your bucket contains the following objects:\n");
    let contents = "";

    while (isTruncated) {
      const { Contents, IsTruncated, NextContinuationToken } =
        await client.send(command);
      const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
      contents += contentsList + "\n";
      isTruncated = IsTruncated;
      command.input.ContinuationToken = NextContinuationToken;
    }
    console.log(contents);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun listBucketObjects(bucketName: String) {
    val request =
        ListObjectsRequest {
            bucket = bucketName
```

```
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.listObjects(request)
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The object is ${myObject.size?.let { calKb(it) }} KBs")
            println("The owner is ${myObject.owner}")
        }
    }
}

private fun calKb(intValue: Long): Long = intValue / 1024
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Listez les objets dans un compartiment.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        try {
            $contents = $this->s3client->listObjectsV2([
                'Bucket' => $this->bucketName,
            ]);
            echo "The contents of your bucket are: \n";
            foreach ($contents['Contents'] as $content) {
                echo $content['Key'] . "\n";
            }
        } catch (Exception $exception) {
```

```
            echo "Failed to list objects in $this->bucketName with error: " .
$exception->getMessage();
            exit("Please fix error with listing objects before continuing.");
        }
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for PHP API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    @staticmethod
    def list(bucket, prefix=None):
        """
        Lists the objects in a bucket, optionally filtered by a prefix.

        :param bucket: The bucket to query. This is a Boto3 Bucket resource.
        :param prefix: When specified, only objects that start with this prefix
 are listed.
        :return: The list of objects.
```

```
        """
        try:
            if not prefix:
                objects = list(bucket.objects.all())
            else:
                objects = list(bucket.objects.filter(Prefix=prefix))
            logger.info(
                "Got objects %s from bucket '%s'", [o.key for o in objects],
bucket.name
            )
        except ClientError:
            logger.exception("Couldn't get objects for bucket '%s'.",
bucket.name)
            raise
        else:
            return objects
```

- Pour plus de détails sur l'API, reportez-vous à la section ListObjectsV2 dans le manuel de référence de l'API AWS SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket actions.
class BucketListObjectsWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
  def initialize(bucket)
    @bucket = bucket
```

```
  end

  # Lists object in a bucket.
  #
  # @param max_objects [Integer] The maximum number of objects to list.
  # @return [Integer] The number of objects listed.
  def list_objects(max_objects)
    count = 0
    puts "The objects in #{@bucket.name} are:"
    @bucket.objects.each do |obj|
      puts "\t#{obj.key}"
      count += 1
      break if count == max_objects
    end
    count
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't list objects in bucket #{bucket.name}. Here's why:
 #{e.message}"
    0
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"

  wrapper = BucketListObjectsWrapper.new(Aws::S3::Bucket.new(bucket_name))
  count = wrapper.list_objects(25)
  puts "Listed #{count} objects."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, voir ListObjectsV2 dans le manuel de référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn list_objects(client: &Client, bucket: &str) -> Result<(), Error> {
    let mut response = client
        .list_objects_v2()
        .bucket(bucket.to_owned())
        .max_keys(10) // In this example, go 10 at a time.
        .into_paginator()
        .send();

    while let Some(result) = response.next().await {
        match result {
            Ok(output) => {
                for object in output.contents() {
                    println!(" - {}", object.key().unwrap_or("Unknown"));
                }
            }
            Err(err) => {
                eprintln!("{err:?}")
            }
        }
    }

    Ok(())
}
```

- Pour plus de détails sur l'API, voir la ListObjectsversion V2 dans le AWS SDK pour la référence de l'API Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   TRY.
       oo_result = lo_s3->listobjectsv2(         " oo_result is returned for
testing purposes. "
         iv_bucket = iv_bucket_name
       ).
       MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
     CATCH /aws1/cx_s3_nosuchbucket.
       MESSAGE 'Bucket does not exist.' TYPE 'E'.
   ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous à la section ListObjectsV2 du AWS SDK pour la référence de l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d’être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public func listBucketFiles(bucket: String) async throws -> [String] {
        let input = ListObjectsV2Input(
            bucket: bucket
        )
        let output = try await client.listObjectsV2(input: input)
        var names: [String] = []

        guard let objList = output.contents else {
            return []
        }

        for obj in objList {
            if let objName = obj.key {
                names.append(objName)
            }
        }

        return names
    }
```

- Pour plus de détails sur l'API, consultez la section ListObjectsV2 dans le AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketAccelerateConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketAccelerateConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// Amazon Simple Storage Service (Amazon S3) Transfer Acceleration is a
/// bucket-level feature that enables you to perform faster data transfers
/// to Amazon S3. This example shows how to configure Transfer
/// Acceleration.
/// </summary>
public class TransferAcceleration
{
    /// <summary>
    /// The main method initializes the client object and sets the
    /// Amazon Simple Storage Service (Amazon S3) bucket name before
    /// calling EnableAccelerationAsync.
    /// </summary>
    public static async Task Main()
    {
        var s3Client = new AmazonS3Client();
        const string bucketName = "doc-example-bucket";

        await EnableAccelerationAsync(s3Client, bucketName);
    }

    /// <summary>
    /// This method sets the configuration to enable transfer acceleration
    /// for the bucket referred to in the bucketName parameter.
    /// </summary>
    /// <param name="client">An Amazon S3 client used to enable the
    /// acceleration on an Amazon S3 bucket.</param>
```

```
        /// <param name="bucketName">The name of the Amazon S3 bucket for which
the
        /// method will be enabling acceleration.</param>
        private static async Task EnableAccelerationAsync(AmazonS3Client client,
string bucketName)
        {
            try
            {
                var putRequest = new PutBucketAccelerateConfigurationRequest
                {
                    BucketName = bucketName,
                    AccelerateConfiguration = new AccelerateConfiguration
                    {
                        Status = BucketAccelerateStatus.Enabled,
                    },
                };
                await client.PutBucketAccelerateConfigurationAsync(putRequest);

                var getRequest = new GetBucketAccelerateConfigurationRequest
                {
                    BucketName = bucketName,
                };
                var response = await
client.GetBucketAccelerateConfigurationAsync(getRequest);

                Console.WriteLine($"Acceleration state = '{response.Status}' ");
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error occurred. Message:'{ex.Message}' when
setting transfer acceleration");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAccelerateConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour définir la configuration accélérée d'un bucket

L'`put-bucket-accelerate-configuration`exemple suivant active la configuration d'accélération pour le compartiment spécifié.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket my-bucket \
    --accelerate-configuration Status=Enabled
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketAccelerateConfigurationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande active l'accélération du transfert pour le compartiment S3 donné.

```
$statusVal = New-Object Amazon.S3.BucketAccelerateStatus('Enabled')
Write-S3BucketAccelerateConfiguration -BucketName 's3testbucket' -
AccelerateConfiguration_Status $statusVal
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAccelerateConfigurationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketAcl`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketAcl`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Gérer les listes de contrôle d'accès (ACL)

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Creates an Amazon S3 bucket with an ACL to control access to the
        /// bucket and the objects stored in it.
        /// </summary>
        /// <param name="client">The initialized client object used to create
        /// an Amazon S3 bucket, with an ACL applied to the bucket.
        /// </param>
        /// <param name="region">The AWS Region where the bucket will be
created.</param>
        /// <param name="newBucketName">The name of the bucket to create.</param>
        /// <returns>A boolean value indicating success or failure.</returns>
        public static async Task<bool> CreateBucketUseCannedACLAsync(IAmazonS3
client, S3Region region, string newBucketName)
        {
            try
            {
                // Create a new Amazon S3 bucket with Canned ACL.
                var putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
                    BucketRegion = region,
                    CannedACL = S3CannedACL.LogDeliveryWrite,
                };
```

```
                PutBucketResponse putBucketResponse = await
client.PutBucketAsync(putBucketRequest);

                return putBucketResponse.HttpStatusCode ==
System.Net.HttpStatusCode.OK;
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Amazon S3 error: {ex.Message}");
            }

            return false;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::putBucketAcl(const Aws::String &bucketName, const Aws::String
 &ownerID,
                              const Aws::String &granteePermission,
                              const Aws::String &granteeType, const Aws::String
 &granteeID,
                              const Aws::String &granteeEmailAddress,
                              const Aws::String &granteeURI, const
 Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::Owner owner;
    owner.SetID(ownerID);
```

```
Aws::S3::Model::Grantee grantee;
grantee.SetType(setGranteeType(granteeType));

if (!granteeEmailAddress.empty()) {
    grantee.SetEmailAddress(granteeEmailAddress);
}

if (!granteeID.empty()) {
    grantee.SetID(granteeID);
}

if (!granteeURI.empty()) {
    grantee.SetURI(granteeURI);
}

Aws::S3::Model::Grant grant;
grant.SetGrantee(grantee);
grant.SetPermission(setGranteePermission(granteePermission));

Aws::Vector<Aws::S3::Model::Grant> grants;
grants.push_back(grant);

Aws::S3::Model::AccessControlPolicy acp;
acp.SetOwner(owner);
acp.SetGrants(grants);

Aws::S3::Model::PutBucketAclRequest request;
request.SetAccessControlPolicy(acp);
request.SetBucket(bucketName);

Aws::S3::Model::PutBucketAclOutcome outcome =
        s3Client.PutBucketAcl(request);

if (!outcome.IsSuccess()) {
    const Aws::S3::S3Error &error = outcome.GetError();

    std::cerr << "Error: putBucketAcl: " << error.GetExceptionName()
              << " - " << error.GetMessage() << std::endl;
} else {
    std::cout << "Successfully added an ACL to the bucket '" << bucketName
              << "'." << std::endl;
}
```

```
    return outcome.IsSuccess();
}

//! Routine which converts a human-readable string to a built-in type
 enumeration.
/*!
 \param access: Human readable string.
 \return Permission: A Permission enum.
*/

Aws::S3::Model::Permission setGranteePermission(const Aws::String &access) {
    if (access == "FULL_CONTROL")
        return Aws::S3::Model::Permission::FULL_CONTROL;
    if (access == "WRITE")
        return Aws::S3::Model::Permission::WRITE;
    if (access == "READ")
        return Aws::S3::Model::Permission::READ;
    if (access == "WRITE_ACP")
        return Aws::S3::Model::Permission::WRITE_ACP;
    if (access == "READ_ACP")
        return Aws::S3::Model::Permission::READ_ACP;
    return Aws::S3::Model::Permission::NOT_SET;
}

//! Routine which converts a human-readable string to a built-in type
 enumeration.
/*!
 \param type: Human readable string.
 \return Type: Type enumeration
*/

Aws::S3::Model::Type setGranteeType(const Aws::String &type) {
    if (type == "Amazon customer by email")
        return Aws::S3::Model::Type::AmazonCustomerByEmail;
    if (type == "Canonical user")
        return Aws::S3::Model::Type::CanonicalUser;
    if (type == "Group")
        return Aws::S3::Model::Type::Group;
    return Aws::S3::Model::Type::NOT_SET;
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Cet exemple accorde `full control` l'autorisation à deux AWS utilisateurs (user1@example.com et user2@example.com) et `read` l'autorisation à tout le monde :

```
aws s3api put-bucket-acl --bucket MyBucket --grant-full-control
 emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read
 uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Consultez http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html pour plus de détails sur les ACL personnalisées (les commandes ACL s3api, par exemple`put-bucket-acl`, utilisent la même notation d'argument abrégée).

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AccessControlPolicy;
import software.amazon.awssdk.services.s3.model.Grant;
import software.amazon.awssdk.services.s3.model.Permission;
import software.amazon.awssdk.services.s3.model.PutBucketAclRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Type;
```

```
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SetAcl {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <bucketName> <id>\s

                Where:
                  bucketName - The Amazon S3 bucket to grant permissions on.\s
                  id - The ID of the owner of this bucket (you can get this value
 from the AWS Management Console).
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String id = args[1];
        System.out.format("Setting access \n");
        System.out.println(" in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        setBucketAcl(s3, bucketName, id);
        System.out.println("Done!");
        s3.close();
    }
```

```
    public static void setBucketAcl(S3Client s3, String bucketName, String id) {
        try {
            Grant ownerGrant = Grant.builder()
                    .grantee(builder -> builder.id(id)
                            .type(Type.CANONICAL_USER))
                    .permission(Permission.FULL_CONTROL)
                    .build();

            List<Grant> grantList2 = new ArrayList<>();
            grantList2.add(ownerGrant);

            AccessControlPolicy acl = AccessControlPolicy.builder()
                    .owner(builder -> builder.id(id))
                    .grants(grantList2)
                    .build();

            PutBucketAclRequest putAclReq = PutBucketAclRequest.builder()
                    .bucket(bucketName)
                    .accessControlPolicy(acl)
                    .build();

            s3.putBucketAcl(putAclReq);

        } catch (S3Exception e) {
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Effectuez une commande PUT pour l'ACL du compartiment.

```
import { PutBucketAclCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Most Amazon S3 use cases don't require the use of access control lists (ACLs).
// We recommend that you disable ACLs, except in unusual circumstances where
// you need to control access for each object individually.
// Consider a policy instead. For more information see https://
docs.aws.amazon.com/AmazonS3/latest/userguide/bucket-policies.html.
export const main = async () => {
  // Grant a user READ access to a bucket.
  const command = new PutBucketAclCommand({
    Bucket: "test-bucket",
    AccessControlPolicy: {
      Grants: [
        {
          Grantee: {
            // The canonical ID of the user. This ID is an obfuscated form of
 your AWS account number.
            // It's unique to Amazon S3 and can't be found elsewhere.
            // For more information, see https://docs.aws.amazon.com/AmazonS3/
latest/userguide/finding-canonical-user-id.html.
            ID: "canonical-id-1",
            Type: "CanonicalUser",
          },
          // One of FULL_CONTROL | READ | WRITE | READ_ACP | WRITE_ACP
          // https://docs.aws.amazon.com/AmazonS3/latest/API/
API_Grant.html#AmazonS3-Type-Grant-Permission
          Permission: "FULL_CONTROL",
        },
      ],
```

```
      Owner: {
        ID: "canonical-id-2",
      },
    },
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun setBucketAcl(
    bucketName: String,
    idVal: String,
) {
    val myGrant =
        Grantee {
            id = idVal
            type = Type.CanonicalUser
        }

    val ownerGrant =
```

```
        Grant {
            grantee = myGrant
            permission = Permission.FullControl
        }

    val grantList = mutableListOf<Grant>()
    grantList.add(ownerGrant)

    val ownerOb =
        Owner {
            id = idVal
        }

    val acl =
        AccessControlPolicy {
            owner = ownerOb
            grants = grantList
        }

    val request =
        PutBucketAclRequest {
            bucket = bucketName
            accessControlPolicy = acl
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.putBucketAcl(request)
        println("An ACL was successfully set on $bucketName")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketAclà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def grant_log_delivery_access(self):
        """
        Grant the AWS Log Delivery group write access to the bucket so that
        Amazon S3 can deliver access logs to the bucket. This is the only
 recommended
        use of an S3 bucket ACL.
        """
        try:
            acl = self.bucket.Acl()
            # Putting an ACL overwrites the existing ACL. If you want to preserve
            # existing grants, append new grants to the list of existing grants.
            grants = acl.grants if acl.grants else []
            grants.append(
                {
                    "Grantee": {
                        "Type": "Group",
                        "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery",
                    },
                    "Permission": "WRITE",
```

```
                }
            )
            acl.put(AccessControlPolicy={"Grants": grants, "Owner": acl.owner})
            logger.info("Granted log delivery access to bucket '%s'",
self.bucket.name)
        except ClientError:
            logger.exception("Couldn't add ACL to bucket '%s'.",
self.bucket.name)
            raise
```

- Pour plus de détails sur l'API, consultez PutBucketAcllе AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketCors`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketCors`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Add CORS configuration to the Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used
        /// to apply the CORS configuration to an Amazon S3 bucket.</param>
        /// <param name="configuration">The CORS configuration to apply.</param>
```

```
        private static async Task PutCORSConfigurationAsync(AmazonS3Client
client, CORSConfiguration configuration)
        {
            PutCORSConfigurationRequest request = new
PutCORSConfigurationRequest()
            {
                BucketName = BucketName,
                Configuration = configuration,
            };

            _ = await client.PutCORSConfigurationAsync(request);
        }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

L'exemple suivant active PUTPOST, et les DELETE demandes provenant de www.example.com, et active les GET demandes provenant de n'importe quel domaine :

```
aws s3api put-bucket-cors --bucket MyBucket --cors-configuration file://cors.json

cors.json:
{
  "CORSRules": [
    {
      "AllowedOrigins": ["http://www.example.com"],
      "AllowedHeaders": ["*"],
      "AllowedMethods": ["PUT", "POST", "DELETE"],
      "MaxAgeSeconds": 3000,
      "ExposeHeaders": ["x-amz-server-side-encryption"]
    },
    {
      "AllowedOrigins": ["*"],
      "AllowedHeaders": ["Authorization"],
      "AllowedMethods": ["GET"],
      "MaxAgeSeconds": 3000
```

```
    }
  ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import java.util.ArrayList;
import java.util.List;
import software.amazon.awssdk.services.s3.model.GetBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.GetBucketCorsResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketCorsRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.CORSRule;
import software.amazon.awssdk.services.s3.model.CORSConfiguration;
import software.amazon.awssdk.services.s3.model.PutBucketCorsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class S3Cors {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:
                    <bucketName> <accountId>\s

                Where:
                    bucketName - The Amazon S3 bucket to upload an object into.
                    accountId - The id of the account that owns the Amazon S3
bucket.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String accountId = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        setCorsInformation(s3, bucketName, accountId);
        getBucketCorsInformation(s3, bucketName, accountId);
        deleteBucketCorsInformation(s3, bucketName, accountId);
        s3.close();
    }

    public static void deleteBucketCorsInformation(S3Client s3, String
bucketName, String accountId) {
        try {
            DeleteBucketCorsRequest bucketCorsRequest =
DeleteBucketCorsRequest.builder()
                    .bucket(bucketName)
                    .expectedBucketOwner(accountId)
                    .build();

            s3.deleteBucketCors(bucketCorsRequest);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
   public static void getBucketCorsInformation(S3Client s3, String bucketName,
String accountId) {
       try {
           GetBucketCorsRequest bucketCorsRequest =
GetBucketCorsRequest.builder()
                   .bucket(bucketName)
                   .expectedBucketOwner(accountId)
                   .build();

           GetBucketCorsResponse corsResponse =
s3.getBucketCors(bucketCorsRequest);
           List<CORSRule> corsRules = corsResponse.corsRules();
           for (CORSRule rule : corsRules) {
               System.out.println("allowOrigins: " + rule.allowedOrigins());
               System.out.println("AllowedMethod: " + rule.allowedMethods());
           }

       } catch (S3Exception e) {

           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void setCorsInformation(S3Client s3, String bucketName, String
accountId) {
       List<String> allowMethods = new ArrayList<>();
       allowMethods.add("PUT");
       allowMethods.add("POST");
       allowMethods.add("DELETE");

       List<String> allowOrigins = new ArrayList<>();
       allowOrigins.add("http://example.com");
       try {
           // Define CORS rules.
           CORSRule corsRule = CORSRule.builder()
                   .allowedMethods(allowMethods)
                   .allowedOrigins(allowOrigins)
                   .build();

           List<CORSRule> corsRules = new ArrayList<>();
           corsRules.add(corsRule);
           CORSConfiguration configuration = CORSConfiguration.builder()
```

```
                    .corsRules(corsRules)
                    .build();

            PutBucketCorsRequest putBucketCorsRequest =
 PutBucketCorsRequest.builder()
                    .bucket(bucketName)
                    .corsConfiguration(configuration)
                    .expectedBucketOwner(accountId)
                    .build();

            s3.putBucketCors(putBucketCorsRequest);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Ajoutez une règle CORS.

```
import { PutBucketCorsCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// By default, Amazon S3 doesn't allow cross-origin requests. Use this command
// to explicitly allow cross-origin requests.
```

```
export const main = async () => {
  const command = new PutBucketCorsCommand({
    Bucket: "test-bucket",
    CORSConfiguration: {
      CORSRules: [
        {
          // Allow all headers to be sent to this bucket.
          AllowedHeaders: ["*"],
          // Allow only GET and PUT methods to be sent to this bucket.
          AllowedMethods: ["GET", "PUT"],
          // Allow only requests from the specified origin.
          AllowedOrigins: ["https://www.example.com"],
          // Allow the entity tag (ETag) header to be returned in the response.
 The ETag header
          // The entity tag represents a specific version of the object. The ETag
 reflects
          // changes only to the contents of an object, not its metadata.
          ExposeHeaders: ["ETag"],
          // How long the requesting browser should cache the preflight response.
 After
          // this time, the preflight request will have to be made again.
          MaxAgeSeconds: 3600,
        },
      ],
    },
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def put_cors(self, cors_rules):
        """
        Apply CORS rules to the bucket. CORS rules specify the HTTP actions that
 are
        allowed from other domains.

        :param cors_rules: The CORS rules to apply.
        """
        try:
            self.bucket.Cors().put(CORSConfiguration={"CORSRules": cors_rules})
            logger.info(
                "Put CORS rules %s for bucket '%s'.", cors_rules,
 self.bucket.name
            )
        except ClientError:
            logger.exception("Couldn't put CORS rules for bucket %s.",
 self.bucket.name)
            raise
```

- Pour plus de détails sur l'API, consultez PutBucketCorsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket CORS configuration.
class BucketCorsWrapper
  attr_reader :bucket_cors

  # @param bucket_cors [Aws::S3::BucketCors] A bucket CORS object configured with
 an existing bucket.
  def initialize(bucket_cors)
    @bucket_cors = bucket_cors
  end

  # Sets CORS rules on a bucket.
  #
  # @param allowed_methods [Array<String>] The types of HTTP requests to allow.
  # @param allowed_origins [Array<String>] The origins to allow.
  # @returns [Boolean] True if the CORS rules were set; otherwise, false.
  def set_cors(allowed_methods, allowed_origins)
    @bucket_cors.put(
      cors_configuration: {
        cors_rules: [
          {
            allowed_methods: allowed_methods,
            allowed_origins: allowed_origins,
            allowed_headers: %w[*],
            max_age_seconds: 3600
```

```
          }
        ]
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set CORS rules for #{@bucket_cors.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- Pour plus de détails sur l'API, reportez-vous PutBucketCorsà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketEncryption`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketEncryption`.

CLI

AWS CLI

Pour configurer le chiffrement côté serveur pour un bucket

L'`put-bucket-encryption`exemple suivant définit le chiffrement AES256 comme valeur par défaut pour le compartiment spécifié.

```
aws s3api put-bucket-encryption \
    --bucket my-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketEncryptionà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande active le chiffrement par défaut côté serveur AES256 avec les clés gérées Amazon S3 (SSE-S3) sur le compartiment donné.

```
$Encryptionconfig = @{ServerSideEncryptionByDefault =
 @{ServerSideEncryptionAlgorithm = "AES256"}}
Set-S3BucketEncryption -BucketName 's3testbucket' -
ServerSideEncryptionConfiguration_ServerSideEncryptionRule $Encryptionconfig
```

- Pour plus de détails sur l'API, reportez-vous PutBucketEncryptionà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketLifecycleConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketLifecycleConfiguration`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Supprimer les téléchargements partitionnés incomplets
- Utiliser les objets soumis au contrôle de version

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Adds lifecycle configuration information to the S3 bucket named in
        /// the bucketName parameter.
        /// </summary>
        /// <param name="client">The S3 client used to call the
        /// PutLifecycleConfigurationAsync method.</param>
        /// <param name="bucketName">A string representing the S3 bucket to
        /// which configuration information will be added.</param>
        /// <param name="configuration">A LifecycleConfiguration object that
        /// will be applied to the S3 bucket.</param>
        public static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
string bucketName, LifecycleConfiguration configuration)
        {
            var request = new PutLifecycleConfigurationRequest()
            {
                BucketName = bucketName,
                Configuration = configuration,
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketLifecycleConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

La commande suivante applique une configuration de cycle de vie à un compartiment nommé `my-bucket` :

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-
configuration  file://lifecycle.json
```

Le fichier `lifecycle.json` est un document JSON situé dans le dossier actuel qui définit deux règles :

```
{
    "Rules": [
        {
            "ID": "Move rotated logs to Glacier",
            "Prefix": "rotated/",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Date": "2015-11-10T00:00:00.000Z",
                    "StorageClass": "GLACIER"
                }
            ]
        },
        {
            "Status": "Enabled",
            "Prefix": "",
            "NoncurrentVersionTransitions": [
                {
                    "NoncurrentDays": 2,
                    "StorageClass": "GLACIER"
                }
            ],
            "ID": "Move old versions to Glacier"
        }
    ]
}
```

La première règle déplace les fichiers avec le préfixe `rotated` vers Glacier à la date spécifiée. La deuxième règle déplace les anciennes versions d'objets vers Glacier lorsqu'elles

ne sont plus actuelles. Pour plus d'informations sur les formats d'horodatage acceptables, consultez la section Spécification des valeurs de paramètres dans le guide de l'utilisateur de la AWS CLI.

- Pour plus de détails sur l'API, reportez-vous PutBucketLifecycleConfigurationà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.Transition;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationRequest;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationResponse;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleRequest;
import software.amazon.awssdk.services.s3.model.TransitionStorageClass;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.ExpirationStatus;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import
 software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class LifecycleConfiguration {
        public static void main(String[] args) {
                final String usage = """

                                Usage:
                                  <bucketName> <accountId>\s

                                Where:
                                  bucketName - The Amazon Simple Storage Service
 (Amazon S3) bucket to upload an object into.
                                  accountId - The id of the account that owns the
 Amazon S3 bucket.
                                """;

                if (args.length != 2) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String bucketName = args[0];
                String accountId = args[1];
                Region region = Region.US_EAST_1;
                S3Client s3 = S3Client.builder()
                                .region(region)
                                .build();

                setLifecycleConfig(s3, bucketName, accountId);
                getLifecycleConfig(s3, bucketName, accountId);
                deleteLifecycleConfig(s3, bucketName, accountId);
                System.out.println("You have successfully created, updated, and
 deleted a Lifecycle configuration");
                s3.close();
        }

        public static void setLifecycleConfig(S3Client s3, String bucketName,
 String accountId) {
                try {
                        // Create a rule to archive objects with the
"glacierobjects/" prefix to Amazon
                        // S3 Glacier.
```

```
                        LifecycleRuleFilter ruleFilter =
LifecycleRuleFilter.builder()
                                        .prefix("glacierobjects/")
                                        .build();

                        Transition transition = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
                                        .days(0)
                                        .build();

                        LifecycleRule rule1 = LifecycleRule.builder()
                                        .id("Archive immediately rule")
                                        .filter(ruleFilter)
                                        .transitions(transition)
                                        .status(ExpirationStatus.ENABLED)
                                        .build();

                        // Create a second rule.
                        Transition transition2 = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
                                        .days(0)
                                        .build();

                        List<Transition> transitionList = new ArrayList<>();
                        transitionList.add(transition2);

                        LifecycleRuleFilter ruleFilter2 =
LifecycleRuleFilter.builder()
                                        .prefix("glacierobjects/")
                                        .build();

                        LifecycleRule rule2 = LifecycleRule.builder()
                                        .id("Archive and then delete rule")
                                        .filter(ruleFilter2)
                                        .transitions(transitionList)
                                        .status(ExpirationStatus.ENABLED)
                                        .build();

                        // Add the LifecycleRule objects to an ArrayList.
                        ArrayList<LifecycleRule> ruleList = new ArrayList<>();
                        ruleList.add(rule1);
                        ruleList.add(rule2);
```

```
                        BucketLifecycleConfiguration lifecycleConfiguration =
BucketLifecycleConfiguration.builder()
                                        .rules(ruleList)
                                        .build();

                        PutBucketLifecycleConfigurationRequest
putBucketLifecycleConfigurationRequest = PutBucketLifecycleConfigurationRequest
                                        .builder()
                                        .bucket(bucketName)

.lifecycleConfiguration(lifecycleConfiguration)
                                        .expectedBucketOwner(accountId)
                                        .build();


s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);

                } catch (S3Exception e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }

        // Retrieve the configuration and add a new rule.
        public static void getLifecycleConfig(S3Client s3, String bucketName,
String accountId) {
                try {
                        GetBucketLifecycleConfigurationRequest
getBucketLifecycleConfigurationRequest = GetBucketLifecycleConfigurationRequest
                                        .builder()
                                        .bucket(bucketName)
                                        .expectedBucketOwner(accountId)
                                        .build();

                        GetBucketLifecycleConfigurationResponse response = s3

.getBucketLifecycleConfiguration(getBucketLifecycleConfigurationRequest);
                        List<LifecycleRule> newList = new ArrayList<>();
                        List<LifecycleRule> rules = response.rules();
                        for (LifecycleRule rule : rules) {
                                newList.add(rule);
                        }
```

```
                        // Add a new rule with both a prefix predicate and a tag
predicate.
                        LifecycleRuleFilter ruleFilter =
LifecycleRuleFilter.builder()
                                        .prefix("YearlyDocuments/")
                                        .build();

                        Transition transition = Transition.builder()

.storageClass(TransitionStorageClass.GLACIER)
                                        .days(3650)
                                        .build();

                        LifecycleRule rule1 = LifecycleRule.builder()
                                        .id("NewRule")
                                        .filter(ruleFilter)
                                        .transitions(transition)
                                        .status(ExpirationStatus.ENABLED)
                                        .build();

                        // Add the new rule to the list.
                        newList.add(rule1);
                        BucketLifecycleConfiguration lifecycleConfiguration =
BucketLifecycleConfiguration.builder()
                                        .rules(newList)
                                        .build();

                        PutBucketLifecycleConfigurationRequest
putBucketLifecycleConfigurationRequest = PutBucketLifecycleConfigurationRequest
                                        .builder()
                                        .bucket(bucketName)

.lifecycleConfiguration(lifecycleConfiguration)
                                        .expectedBucketOwner(accountId)
                                        .build();


s3.putBucketLifecycleConfiguration(putBucketLifecycleConfigurationRequest);

                } catch (S3Exception e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
```

```
        // Delete the configuration from the Amazon S3 bucket.
        public static void deleteLifecycleConfig(S3Client s3, String bucketName,
 String accountId) {
                try {
                        DeleteBucketLifecycleRequest deleteBucketLifecycleRequest
 = DeleteBucketLifecycleRequest
                                        .builder()
                                        .bucket(bucketName)
                                        .expectedBucketOwner(accountId)
                                        .build();

                        s3.deleteBucketLifecycle(deleteBucketLifecycleRequest);

                } catch (S3Exception e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketLifecycleConfigurationà la section Référence des AWS SDK for Java 2.x API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
```

```
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def put_lifecycle_configuration(self, lifecycle_rules):
        """
        Apply a lifecycle configuration to the bucket. The lifecycle
configuration can
        be used to archive or delete the objects in the bucket according to
specified
        parameters, such as a number of days.

        :param lifecycle_rules: The lifecycle rules to apply.
        """
        try:
            self.bucket.LifecycleConfiguration().put(
                LifecycleConfiguration={"Rules": lifecycle_rules}
            )
            logger.info(
                "Put lifecycle rules %s for bucket '%s'.",
                lifecycle_rules,
                self.bucket.name,
            )
        except ClientError:
            logger.exception(
                "Couldn't put lifecycle rules for bucket '%s'.", self.bucket.name
            )
            raise
```

- Pour plus de détails sur l'API, consultez PutBucketLifecycleConfigurationle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **PutBucketLogging** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketLogging`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();

        string bucketName = _configuration["BucketName"];
        string logBucketName = _configuration["LogBucketName"];
        string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
        string accountId = _configuration["AccountId"];

        // If the AWS Region defined for your default user is different
```

```
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
            IAmazonS3 client = new AmazonS3Client();

            try
            {
                // Update bucket policy for target bucket to allow delivery of
logs to it.
                await SetBucketPolicyToAllowLogDelivery(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix,
                    accountId);

                // Enable logging on the source bucket.
                await EnableLoggingAsync(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error: {e.Message}");
            }
        }

        /// <summary>
        /// This method grants appropriate permissions for logging to the
        /// Amazon S3 bucket where the logs will be stored.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
used
        /// to apply the bucket policy.</param>
        /// <param name="sourceBucketName">The name of the source bucket.</param>
        /// <param name="logBucketName">The name of the bucket where logging
        /// information will be stored.</param>
        /// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
        /// <param name="accountId">The account id of the account where the
source bucket exists.</param>
        /// <returns>Async task.</returns>
```

```
        public static async Task SetBucketPolicyToAllowLogDelivery(
            IAmazonS3 client,
            string sourceBucketName,
            string logBucketName,
            string logPrefix,
            string accountId)
        {
            var resourceArn = @"""arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @"*""";

            var newPolicy = @"{
                                ""Statement"":[{
                                ""Sid"": ""S3ServerAccessLogsPolicy"",
                                ""Effect"": ""Allow"",
                                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                                ""Action"": [""s3:PutObject""],
                                ""Resource"": [" + resourceArn + @"],
                                ""Condition"": {
                                ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @""" },
                                ""StringEquals"": { ""aws:SourceAccount"": """ +
accountId + @""" }
                                        }
                                    }]
                                }";
            Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
            Console.WriteLine(newPolicy);

            PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
            {
                BucketName = logBucketName,
                Policy = newPolicy,
            };
            await client.PutBucketPolicyAsync(putRequest);
            Console.WriteLine("Policy applied.");
        }

        /// <summary>
        /// This method enables logging for an Amazon S3 bucket. Logs will be
stored
        /// in the bucket you selected for logging. Selected prefix
        /// will be prepended to each log object.
```

```
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
 used
        /// to configure and apply logging to the selected Amazon S3 bucket.</
param>
        /// <param name="bucketName">The name of the Amazon S3 bucket for which
 you
        /// wish to enable logging.</param>
        /// <param name="logBucketName">The name of the Amazon S3 bucket where
 logging
        /// information will be stored.</param>
        /// <param name="logObjectKeyPrefix">The prefix to prepend to each
        /// object key.</param>
        /// <returns>Async task.</returns>
        public static async Task EnableLoggingAsync(
            IAmazonS3 client,
            string bucketName,
            string logBucketName,
            string logObjectKeyPrefix)
        {
            Console.WriteLine($"Enabling logging for bucket {bucketName}.");
            var loggingConfig = new S3BucketLoggingConfig
            {
                TargetBucketName = logBucketName,
                TargetPrefix = logObjectKeyPrefix,
            };

            var putBucketLoggingRequest = new PutBucketLoggingRequest
            {
                BucketName = bucketName,
                LoggingConfig = loggingConfig,
            };
            await client.PutBucketLoggingAsync(putBucketLoggingRequest);
            Console.WriteLine($"Logging enabled.");
        }

        /// <summary>
        /// Loads configuration from settings files.
        /// </summary>
        public static void LoadConfig()
        {
            _configuration = new ConfigurationBuilder()
                .SetBasePath(Directory.GetCurrentDirectory())
                .AddJsonFile("settings.json") // Load settings from .json file.
```

```
                .AddJsonFile("settings.local.json", true) // Optionally, load
 local settings.
                .Build();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketLoggingà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Exemple 1 : pour définir la journalisation des politiques relatives aux compartiments

L'`put-bucket-logging`exemple suivant définit la politique de journalisation pour MyBucket. Tout d'abord, accordez au service de journalisation l'autorisation principale dans votre politique de compartiment à l'aide de la `put-bucket-policy` commande.

```
aws s3api put-bucket-policy \
    --bucket MyBucket \
    --policy file://policy.json
```

Contenu de `policy.json` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {"Service": "logging.s3.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::MyBucket/Logs/*",
            "Condition": {
                "ArnLike": {"aws:SourceARN": "arn:aws:s3:::SOURCE-BUCKET-NAME"},
                "StringEquals": {"aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"}
            }
        }
    ]
```

```
}
```

Pour appliquer la politique de journalisation, utilisezput-bucket-logging.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

Contenu de logging.json :

```
{
     "LoggingEnabled": {
         "TargetBucket": "MyBucket",
         "TargetPrefix": "Logs/"
     }
 }
```

La put-bucket-policy commande est requise pour accorder s3:PutObject des autorisations au principal du service de journalisation.

Pour plus d'informations, consultez la section Journalisation de l'accès au serveur Amazon S3 dans le guide de l'utilisateur Amazon S3.

Exemple 2 : pour définir une politique de compartiment pour la journalisation de l'accès à un seul utilisateur

L'put-bucket-loggingexemple suivant définit la politique de journalisation pour MyBucket. L' AWS utilisateur bob@example.com aura un contrôle total sur les fichiers journaux, et personne d'autre n'y aura accès. Tout d'abord, accordez l'autorisation S3 avecput-bucket-acl.

```
aws s3api put-bucket-acl \
    --bucket MyBucket \
    --grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery \
    --grant-read-acp URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

Appliquez ensuite la politique de journalisation à l'aide deput-bucket-logging.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

Contenu de `logging.json` :

```
{
    "LoggingEnabled": {
        "TargetBucket": "MyBucket",
        "TargetPrefix": "MyBucketLogs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

la `put-bucket-acl` commande est requise pour accorder au système de livraison de journaux de S3 les autorisations nécessaires (autorisations d'écriture et de lecture).

Pour plus d'informations, consultez la section Journalisation de l'accès au serveur Amazon S3 dans le manuel du développeur Amazon S3.

- Pour plus de détails sur l'API, reportez-vous PutBucketLoggingà la section Référence des AWS CLI commandes.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketNotification`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketNotification`.

CLI

AWS CLI

Applique une configuration de notification à un compartiment nommé `my-bucket` :

```
aws s3api put-bucket-notification --bucket my-bucket --notification-configuration
 file://notification.json
```

Le fichier `notification.json` est un document JSON situé dans le dossier actuel qui indique une rubrique SNS et un type d'événement à surveiller :

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

La rubrique SNS doit être associée à une politique IAM qui autorise Amazon S3 à y publier :

```
{
 "Version": "2008-10-17",
 "Id": "example-ID",
 "Statement": [
  {
   "Sid": "example-statement-ID",
   "Effect": "Allow",
   "Principal": {
     "Service": "s3.amazonaws.com"
   },
   "Action": [
    "SNS:Publish"
   ],
   "Resource": "arn:aws:sns:us-west-2:123456789012:my-bucket",
   "Condition": {
      "ArnLike": {
      "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
    }
   }
  }
 ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cet exemple configure la configuration de la rubrique SNS pour l'événement S3 ObjectRemovedDelete et active les notifications pour le compartiment s3 donné

```
$topic =  [Amazon.S3.Model.TopicConfiguration] @{
  Id = "delete-event"
  Topic = "arn:aws:sns:eu-west-1:123456789012:topic-1"
  Event = [Amazon.S3.EventType]::ObjectRemovedDelete
}

Write-S3BucketNotification -BucketName kt-tools -TopicConfiguration $topic
```

Exemple 2 : Cet exemple active les notifications ObjectCreatedAll pour le bucket donné qui l'envoient à la fonction Lambda.

```
$lambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
  Events = "s3:ObjectCreated:*"
  FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:rdplock"
  Id = "ObjectCreated-Lambda"
  Filter = @{
    S3KeyFilter = @{
      FilterRules = @(
        @{Name="Prefix";Value="dada"}
        @{Name="Suffix";Value=".pem"}
      )
    }
  }
}

Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration
 $lambdaConfig
```

Exemple 3 : Cet exemple crée 2 configurations Lambda différentes sur la base d'un suffixe clé différent et configure les deux en une seule commande.

```
#Lambda Config 1

$firstLambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
```

```
  Events = "s3:ObjectCreated:*"
  FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifynet"
  Id = "ObjectCreated-dada-ps1"
  Filter = @{
    S3KeyFilter = @{
      FilterRules = @(
        @{Name="Prefix";Value="dada"}
        @{Name="Suffix";Value=".ps1"}
      )
    }
  }
}

#Lambda Config 2

$secondlambdaConfig = [Amazon.S3.Model.LambdaFunctionConfiguration] @{
  Events = [Amazon.S3.EventType]::ObjectCreatedAll
  FunctionArn = "arn:aws:lambda:eu-west-1:123456789012:function:verifyssm"
  Id = "ObjectCreated-dada-json"
  Filter = @{
    S3KeyFilter = @{
      FilterRules = @(
        @{Name="Prefix";Value="dada"}
        @{Name="Suffix";Value=".json"}
      )
    }
  }
}

Write-S3BucketNotification -BucketName ssm-editor -LambdaFunctionConfiguration
 $firstLambdaConfig,$secondlambdaConfig
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **PutBucketNotificationConfiguration** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketNotificationConfiguration`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    using System;
    using System.Collections.Generic;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example shows how to enable notifications for an Amazon Simple
    /// Storage Service (Amazon S3) bucket.
    /// </summary>
    public class EnableNotifications
    {
        public static async Task Main()
        {
            const string bucketName = "doc-example-bucket1";
            const string snsTopic = "arn:aws:sns:us-east-2:0123456789ab:bucket-
notify";
            const string sqsQueue = "arn:aws:sqs:us-
east-2:0123456789ab:Example_Queue";

            IAmazonS3 client = new AmazonS3Client(Amazon.RegionEndpoint.USEast2);
            await EnableNotificationAsync(client, bucketName, snsTopic,
 sqsQueue);
        }
```

```
        /// <summary>
        /// This method makes the call to the PutBucketNotificationAsync method.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client used to call
        /// the PutBucketNotificationAsync method.</param>
        /// <param name="bucketName">The name of the bucket for which
        /// notifications will be turned on.</param>
        /// <param name="snsTopic">The ARN for the Amazon Simple Notification
        /// Service (Amazon SNS) topic associated with the S3 bucket.</param>
        /// <param name="sqsQueue">The ARN of the Amazon Simple Queue Service
        /// (Amazon SQS) queue to which notifications will be pushed.</param>
        public static async Task EnableNotificationAsync(
            IAmazonS3 client,
            string bucketName,
            string snsTopic,
            string sqsQueue)
        {
            try
            {
                // The bucket for which we are setting up notifications.
                var request = new PutBucketNotificationRequest()
                {
                    BucketName = bucketName,
                };

                // Defines the topic to use when sending a notification.
                var topicConfig = new TopicConfiguration()
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic,
                };
                request.TopicConfigurations = new List<TopicConfiguration>
                {
                    topicConfig,
                };
                request.QueueConfigurations = new List<QueueConfiguration>
                {
                    new QueueConfiguration()
                    {
                        Events = new List<EventType>
{ EventType.ObjectCreatedPut },
                        Queue = sqsQueue,
                    },
                };
```

```
                // Now apply the notification settings to the bucket.
                PutBucketNotificationResponse response = await
 client.PutBucketNotificationAsync(request);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour activer les notifications spécifiées dans un bucket

L'`put-bucket-notification-configuration`exemple suivant applique une configuration de notification à un compartiment nommé`my-bucket`. Le fichier `notification.json` est un document JSON situé dans le dossier actuel qui spécifie une rubrique SNS et un type d'événement à surveiller.

```
aws s3api put-bucket-notification-configuration \
    --bucket my-bucket \
    --notification-configuration file://notification.json
```

Contenu de `notification.json` :

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-
topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
```

```
        }
    ]
}
```

La rubrique SNS doit être associée à une politique IAM qui autorise Amazon S3 à y publier.

```
{
    "Version": "2008-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "arn:aws:sns:us-west-2:123456789012::s3-notification-
topic",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:my-bucket"
                }
            }
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationConfigurationà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.Event;
import software.amazon.awssdk.services.s3.model.NotificationConfiguration;
import
 software.amazon.awssdk.services.s3.model.PutBucketNotificationConfigurationRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.TopicConfiguration;
import java.util.ArrayList;
import java.util.List;

public class SetBucketEventBridgeNotification {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName>\s

                Where:
                    bucketName - The Amazon S3 bucket.\s
                    topicArn - The Simple Notification Service topic ARN.\s
                    id - An id value used for the topic configuration. This value
 is displayed in the AWS Management Console.\s
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String topicArn = args[1];
        String id = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        setBucketNotification(s3Client, bucketName, topicArn, id);
        s3Client.close();
    }
```

```
    public static void setBucketNotification(S3Client s3Client, String
 bucketName, String topicArn, String id) {
        try {
            List<Event> events = new ArrayList<>();
            events.add(Event.S3_OBJECT_CREATED_PUT);

            TopicConfiguration config = TopicConfiguration.builder()
                    .topicArn(topicArn)
                    .events(events)
                    .id(id)
                    .build();

            List<TopicConfiguration> topics = new ArrayList<>();
            topics.add(config);

            NotificationConfiguration configuration =
 NotificationConfiguration.builder()
                    .topicConfigurations(topics)
                    .build();

            PutBucketNotificationConfigurationRequest configurationRequest =
 PutBucketNotificationConfigurationRequest
                    .builder()
                    .bucket(bucketName)
                    .notificationConfiguration(configuration)
                    .skipDestinationValidation(true)
                    .build();

            // Set the bucket notification configuration.
            s3Client.putBucketNotificationConfiguration(configurationRequest);
            System.out.println("Added bucket " + bucketName + " with EventBridge
 events enabled.");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketNotificationConfigurationà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketPolicy`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketPolicy`.

C++

Kit de développement logiciel (SDK) for C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::putBucketPolicy(const Aws::String &bucketName,
                                 const Aws::String &policyBody,
                                 const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    std::shared_ptr<Aws::StringStream> request_body =
            Aws::MakeShared<Aws::StringStream>("");
    *request_body << policyBody;

    Aws::S3::Model::PutBucketPolicyRequest request;
    request.SetBucket(bucketName);
    request.SetBody(request_body);

    Aws::S3::Model::PutBucketPolicyOutcome outcome =
            s3Client.PutBucketPolicy(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: putBucketPolicy: "
                  << outcome.GetError().GetMessage() << std::endl;
    } else {
        std::cout << "Set the following policy body for the bucket '" <<
                  bucketName << "':" << std::endl << std::endl;
        std::cout << policyBody << std::endl;
```

```
    }

    return outcome.IsSuccess();
}


//! Build a policy JSON string.
/*!
  \param userArn: Aws user Amazon Resource Name (ARN).
      For more information, see https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_identifiers.html#identifiers-arns.
  \param bucketName: Name of a bucket.
  \return String: Policy as JSON string.
*/

Aws::String getPolicyString(const Aws::String &userArn,
                            const Aws::String &bucketName) {
    return
            "{\n"
            "   \"Version\":\"2012-10-17\",\n"
            "   \"Statement\":[\n"
            "       {\n"
            "           \"Sid\": \"1\",\n"
            "           \"Effect\": \"Allow\",\n"
            "           \"Principal\": {\n"
            "               \"AWS\": \""
            + userArn +
            "\"\n""           },\n"
            "           \"Action\": [ \"s3:getObject\" ],\n"
            "           \"Resource\": [ \"arn:aws:s3:::"
            + bucketName +
            "/*\" ]\n"
            "       }\n"
            "   ]\n"
            "}";
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Cet exemple permet à tous les utilisateurs de récupérer n'importe quel objet MyBucketsauf ceux du MySecretFolder. Il accorde `put` également une `delete` autorisation à l'utilisateur root du AWS compte `1234-5678-9012` :

```
aws s3api put-bucket-policy --bucket MyBucket --policy file://policy.json

policy.json:
{
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::MyBucket/*"
      },
      {
         "Effect": "Deny",
         "Principal": "*",
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::MyBucket/MySecretFolder/*"
      },
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:root"
         },
         "Action": [
            "s3:DeleteObject",
            "s3:PutObject"
         ],
         "Resource": "arn:aws:s3:::MyBucket/*"
      }
   ]
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketPolicyRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SetBucketPolicy {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <polFile>

                Where:
                    bucketName - The Amazon S3 bucket to set the policy on.
                    polFile - A JSON file containing the policy (see the Amazon
 S3 Readme for an example).\s
```

```
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String polFile = args[1];
        String policyText = getBucketPolicyFromFile(polFile);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        setPolicy(s3, bucketName, policyText);
        s3.close();
    }

    public static void setPolicy(S3Client s3, String bucketName, String
policyText) {
        System.out.println("Setting policy:");
        System.out.println("----");
        System.out.println(policyText);
        System.out.println("----");
        System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

        try {
            PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
                    .bucket(bucketName)
                    .policy(policyText)
                    .build();

            s3.putBucketPolicy(policyReq);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        System.out.println("Done!");
    }

    // Loads a JSON-formatted policy from a file
```

```
    public static String getBucketPolicyFromFile(String policyFile) {

        StringBuilder fileText = new StringBuilder();
        try {
            List<String> lines = Files.readAllLines(Paths.get(policyFile),
 StandardCharsets.UTF_8);
            for (String line : lines) {
                fileText.append(line);
            }

        } catch (IOException e) {
            System.out.format("Problem reading file: \"%s\"", policyFile);
            System.out.println(e.getMessage());
        }

        try {
            final JsonParser parser = new
 ObjectMapper().getFactory().createParser(fileText.toString());
            while (parser.nextToken() != null) {
            }

        } catch (IOException jpe) {
            jpe.printStackTrace();
        }
        return fileText.toString();
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Ajoutez la politique.

```
import { PutBucketPolicyCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutBucketPolicyCommand({
    Policy: JSON.stringify({
      Version: "2012-10-17",
      Statement: [
        {
          Sid: "AllowGetObject",
          // Allow this particular user to call GetObject on any object in this
 bucket.
          Effect: "Allow",
          Principal: {
            AWS: "arn:aws:iam::ACCOUNT-ID:user/USERNAME",
          },
          Action: "s3:GetObject",
          Resource: "arn:aws:s3:::BUCKET-NAME/*",
        },
      ],
    }),
    // Apply the preceding policy to this bucket.
    Bucket: "BUCKET-NAME",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS SDK for JavaScript API.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def put_policy(self, policy):
        """
        Apply a security policy to the bucket. Policies control users' ability
        to perform specific actions, such as listing the objects in the bucket.

        :param policy: The policy to apply to the bucket.
        """
        try:
            self.bucket.Policy().put(Policy=json.dumps(policy))
            logger.info("Put policy %s for bucket '%s'.", policy,
 self.bucket.name)
        except ClientError:
            logger.exception("Couldn't apply policy to bucket '%s'.",
 self.bucket.name)
            raise
```

- Pour plus de détails sur l'API, consultez PutBucketPolicyle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
# Wraps an Amazon S3 bucket policy.
class BucketPolicyWrapper
  attr_reader :bucket_policy

  # @param bucket_policy [Aws::S3::BucketPolicy] A bucket policy object
 configured with an existing bucket.
  def initialize(bucket_policy)
    @bucket_policy = bucket_policy
  end

  # Sets a policy on a bucket.
  #
  def set_policy(policy)
    @bucket_policy.put(policy: policy)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't set the policy for #{@bucket_policy.bucket.name}. Here's why:
 #{e.message}"
    false
  end

end
```

- Pour plus de détails sur l'API, reportez-vous PutBucketPolicyà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **PutBucketReplication** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliserPutBucketReplication.

CLI

AWS CLI

Pour configurer la réplication pour un compartiment S3

L'put-bucket-replicationexemple suivant applique une configuration de réplication au compartiment S3 spécifié.

```
aws s3api put-bucket-replication \
    --bucket AWSDOC-EXAMPLE-BUCKET1 \
    --replication-configuration file://replication.json
```

Contenu de replication.json :

```
{
    "Role": "arn:aws:iam::123456789012:role/s3-replication-role",
    "Rules": [
        {
            "Status": "Enabled",
            "Priority": 1,
            "DeleteMarkerReplication": { "Status": "Disabled" },
            "Filter" : { "Prefix": ""},
            "Destination": {
                "Bucket": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2"
            }
        }
    ]
}
```

La gestion des versions doit être activée dans le compartiment de destination. Le rôle spécifié doit être autorisé à écrire dans le compartiment de destination et avoir une relation de confiance qui permet à Amazon S3 d'assumer le rôle.

Exemple de politique d'autorisation des rôles :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetReplicationConfiguration",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion",
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET1/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ReplicateObject",
                "s3:ReplicateDelete",
                "s3:ReplicateTags"
            ],
            "Resource": "arn:aws:s3:::AWSDOC-EXAMPLE-BUCKET2/*"
        }
    ]
}
```

Exemple de politique de relation de confiance :

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez Ceci est le titre du sujet dans le guide de l'utilisateur de la console Amazon Simple Storage Service.

- Pour plus de détails sur l'API, reportez-vous PutBucketReplicationà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cet exemple définit une configuration de réplication avec une règle unique permettant de répliquer dans le compartiment « exampletargetbucket » tous les nouveaux objets créés avec le préfixe de nom clé « » dans le compartiment « examplebucket »TaxDocs.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}

Write-S3BucketReplication @params
```

Exemple 2 : Cet exemple définit une configuration de réplication avec plusieurs règles permettant de répliquer dans le compartiment « exampletargetbucket » tous les nouveaux objets créés avec le préfixe de nom de clé « » ou « ». TaxDocs OtherDocs Les préfixes clés ne doivent pas se chevaucher.

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Enabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$rule2 = New-Object Amazon.S3.Model.ReplicationRule
$rule2.ID = "Rule-2"
$rule2.Status = "Enabled"
$rule2.Prefix = "OtherDocs"
$rule2.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1,$rule2
}

Write-S3BucketReplication @params
```

Exemple 3 : Cet exemple met à jour la configuration de réplication sur le compartiment spécifié afin de désactiver la règle contrôlant la réplication des objets portant le préfixe de nom clé « » vers le compartiment TaxDocs « exampletargetbucket ».

```
$rule1 = New-Object Amazon.S3.Model.ReplicationRule
$rule1.ID = "Rule-1"
$rule1.Status = "Disabled"
$rule1.Prefix = "TaxDocs"
$rule1.Destination = @{ BucketArn = "arn:aws:s3:::exampletargetbucket" }

$params = @{
    BucketName = "examplebucket"
    Configuration_Role = "arn:aws:iam::35667example:role/
CrossRegionReplicationRoleForS3"
    Configuration_Rule = $rule1
}
```

```
Write-S3BucketReplication @params
```

- Pour plus de détails sur l'API, reportez-vous PutBucketReplicationà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **PutBucketRequestPayment** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliserPutBucketRequestPayment.

CLI

AWS CLI

Exemple 1 : Pour activer la configuration ``requester pay`` pour un bucket

L'put-bucket-request-paymentexemple suivant active requester pays le bucket spécifié.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

Cette commande ne produit aucun résultat.

Exemple 2 : Pour désactiver la configuration ``requester pay`` pour un bucket

L'put-bucket-request-paymentexemple suivant désactive requester pays le bucket spécifié.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutBucketRequestPaymentà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : met à jour la configuration du paiement des demandes pour le compartiment nommé « mybucket » afin que le téléchargement soit facturé à la personne demandant des téléchargements depuis le compartiment. Par défaut, le propriétaire du bucket paie les téléchargements. Pour rétablir la valeur par défaut du paiement de la demande, utilisez « BucketOwner » pour le paramètre RequestPaymentConfiguration _Payer.

```
Write-S3BucketRequestPayment -BucketName mybucket -
RequestPaymentConfiguration_Payer Requester
```

- Pour plus de détails sur l'API, reportez-vous PutBucketRequestPaymentà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketTagging`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketTagging`.

CLI

AWS CLI

La commande suivante applique une configuration de balisage à un compartiment nommé `my-bucket` :

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

Le fichier `tagging.json` est un document JSON situé dans le dossier actuel qui spécifie les balises :

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
}
```

Ou appliquez une configuration de balisage `my-bucket` directement depuis la ligne de commande :

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- Pour plus de détails sur l'API, reportez-vous PutBucketTaggingà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : Cette commande applique deux balises à un compartiment nommé **`cloudtrail-test-2018`** : une balise avec une clé Stage et une valeur Test, et une balise avec une clé Environment et une valeur Alpha. Pour vérifier que les balises ont été ajoutées au compartiment, exécutez**`Get-S3BucketTagging -BucketName bucket_name`**. Les résultats doivent indiquer les balises que vous avez appliquées au bucket lors de la première commande. Notez que cela **`Write-S3BucketTagging`** remplace la totalité du jeu de balises existant sur un bucket. Pour ajouter ou supprimer des balises individuelles, exécutez les applets de commande Resource Groups et Tagging API, et. **`Add-RGTResourceTag`** **`Remove-RGTResourceTag`** Vous pouvez également utiliser l'éditeur de balises dans la console AWS de gestion pour gérer les balises de compartiment S3.

```
Write-S3BucketTagging -BucketName cloudtrail-test-2018 -TagSet @( @{ Key="Stage";
 Value="Test" }, @{ Key="Environment"; Value="Alpha" } )
```

Exemple 2 : Cette commande dirige un bucket nommé **`cloudtrail-test-2018`** vers l'**`Write-S3BucketTagging`**applet de commande. Il applique les balises Stage:Production

et Department:Finance au bucket. Notez que cela **`Write-S3BucketTagging`** remplace la totalité du jeu de balises existant sur un bucket.

```
Get-S3Bucket -BucketName cloudtrail-test-2018 | Write-S3BucketTagging
 -TagSet @( @{ Key="Stage"; Value="Production" }, @{ Key="Department";
 Value="Finance" } )
```

- Pour plus de détails sur l'API, reportez-vous PutBucketTaggingà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketVersioning`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketVersioning`.

CLI

AWS CLI

La commande suivante active le versionnement sur un bucket nommé `my-bucket` :

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
 Status=Enabled
```

La commande suivante active le versionnement et utilise un code mfa

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-configuration
 Status=Enabled --mfa "SERIAL 123456"
```

- Pour plus de détails sur l'API, reportez-vous PutBucketVersioningà la section Référence des AWS CLI commandes.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande active la gestion des versions pour le compartiment S3 donné.

```
Write-S3BucketVersioning -BucketName 's3testbucket' -VersioningConfig_Status
 Enabled
```

- Pour plus de détails sur l'API, reportez-vous PutBucketVersioningà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutBucketWebsite`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutBucketWebsite`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
                // Put the website configuration.
                PutBucketWebsiteRequest putRequest = new
PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument,
                    },
                };
                PutBucketWebsiteResponse response = await
client.PutBucketWebsiteAsync(putRequest);
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::putWebsiteConfig(const Aws::String &bucketName,
                                  const Aws::String &indexPage, const Aws::String
 &errorPage,
                                  const Aws::S3::S3ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client client(clientConfig);

    Aws::S3::Model::IndexDocument indexDocument;
    indexDocument.SetSuffix(indexPage);

    Aws::S3::Model::ErrorDocument errorDocument;
    errorDocument.SetKey(errorPage);

    Aws::S3::Model::WebsiteConfiguration websiteConfiguration;
    websiteConfiguration.SetIndexDocument(indexDocument);
    websiteConfiguration.SetErrorDocument(errorDocument);

    Aws::S3::Model::PutBucketWebsiteRequest request;
    request.SetBucket(bucketName);
    request.SetWebsiteConfiguration(websiteConfiguration);

    Aws::S3::Model::PutBucketWebsiteOutcome outcome =
            client.PutBucketWebsite(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: PutBucketWebsite: "
                  << outcome.GetError().GetMessage() << std::endl;
    } else {
```

```
        std::cout << "Success: Set website configuration for bucket '"
                  << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Applique une configuration de site Web statique à un compartiment nommé `my-bucket` :

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://
website.json
```

Le fichier `website.json` est un document JSON situé dans le dossier actuel qui indique les pages d'index et d'erreur du site Web :

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.IndexDocument;
import software.amazon.awssdk.services.s3.model.PutBucketWebsiteRequest;
import software.amazon.awssdk.services.s3.model.WebsiteConfiguration;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.regions.Region;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class SetWebsiteConfiguration {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <bucketName> [indexdoc]\s

                Where:
                   bucketName   - The Amazon S3 bucket to set the website
 configuration on.\s
                   indexdoc - The index document, ex. 'index.html'
                              If not specified, 'index.html' will be set.
                """;

        if (args.length != 1) {
            System.out.println(usage);
```

```
            System.exit(1);
        }

        String bucketName = args[0];
        String indexDoc = "index.html";
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        setWebsiteConfig(s3, bucketName, indexDoc);
        s3.close();
    }

    public static void setWebsiteConfig(S3Client s3, String bucketName, String
 indexDoc) {
        try {
            WebsiteConfiguration websiteConfig = WebsiteConfiguration.builder()

 .indexDocument(IndexDocument.builder().suffix(indexDoc).build())
                    .build();

            PutBucketWebsiteRequest pubWebsiteReq =
 PutBucketWebsiteRequest.builder()
                    .bucket(bucketName)
                    .websiteConfiguration(websiteConfig)
                    .build();

            s3.putBucketWebsite(pubWebsiteReq);
            System.out.println("The call was successful");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez la configuration du site web.

```
import { PutBucketWebsiteCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

// Set up a bucket as a static website.
// The bucket needs to be publicly accessible.
export const main = async () => {
  const command = new PutBucketWebsiteCommand({
    Bucket: "test-bucket",
    WebsiteConfiguration: {
      ErrorDocument: {
        // The object key name to use when a 4XX class error occurs.
        Key: "error.html",
      },
      IndexDocument: {
        // A suffix that is appended to a request that is for a directory.
        Suffix: "index.html",
      },
    },
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande active l'hébergement du site Web pour le compartiment donné avec le document d'index « index.html » et le document d'erreur « error.html ».

```
Write-S3BucketWebsite -BucketName 's3testbucket' -
WebsiteConfiguration_IndexDocumentSuffix 'index.html' -
WebsiteConfiguration_ErrorDocument 'error.html'
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS Tools for PowerShell applets de commande.

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps Amazon S3 bucket website actions.
class BucketWebsiteWrapper
  attr_reader :bucket_website

  # @param bucket_website [Aws::S3::BucketWebsite] A bucket website object
 configured with an existing bucket.
  def initialize(bucket_website)
    @bucket_website = bucket_website
```

```
  end

  # Sets a bucket as a static website.
  #
  # @param index_document [String] The name of the index document for the
 website.
  # @param error_document [String] The name of the error document to show for 4XX
 errors.
  # @return [Boolean] True when the bucket is configured as a website; otherwise,
 false.
  def set_website(index_document, error_document)
    @bucket_website.put(
      website_configuration: {
        index_document: { suffix: index_document },
        error_document: { key: error_document }
      }
    )
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't configure #{@bucket_website.bucket.name} as a website. Here's
 why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  index_document = "index.html"
  error_document = "404.html"

  wrapper = BucketWebsiteWrapper.new(Aws::S3::BucketWebsite.new(bucket_name))
  return unless wrapper.set_website(index_document, error_document)

  puts "Successfully configured bucket #{bucket_name} as a static website."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous PutBucketWebsiteà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutObject`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Démarrer avec les compartiments et les objets
- Suivez les chargements et les téléchargements

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        /// <summary>
        /// Shows how to upload a file from the local computer to an Amazon S3
        /// bucket.
        /// </summary>
        /// <param name="client">An initialized Amazon S3 client object.</param>
        /// <param name="bucketName">The Amazon S3 bucket to which the object
        /// will be uploaded.</param>
        /// <param name="objectName">The object to upload.</param>
        /// <param name="filePath">The path, including file name, of the object
        /// on the local computer to upload.</param>
        /// <returns>A boolean value indicating the success or failure of the
        /// upload procedure.</returns>
        public static async Task<bool> UploadFileAsync(
            IAmazonS3 client,
```

```
            string bucketName,
            string objectName,
            string filePath)
        {
            var request = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = objectName,
                FilePath = filePath,
            };

            var response = await client.PutObjectAsync(request);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully uploaded {objectName} to
{bucketName}.");
                return true;
            }
            else
            {
                Console.WriteLine($"Could not upload {objectName} to
{bucketName}.");
                return false;
            }
        }
```

Chargez un objet avec un chiffrement côté serveur.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example shows how to upload an object to an Amazon Simple Storage
    /// Service (Amazon S3) bucket with server-side encryption enabled.
    /// </summary>
    public class ServerSideEncryption
    {
        public static async Task Main()
        {
```

```
            string bucketName = "doc-example-bucket";
            string keyName = "samplefile.txt";

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2.
            IAmazonS3 client = new AmazonS3Client();

            await WritingAnObjectAsync(client, bucketName, keyName);
        }

        /// <summary>
        /// Upload a sample object include a setting for encryption.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
        /// to upload a file and apply server-side encryption.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket where the
        /// encrypted object will reside.</param>
        /// <param name="keyName">The name for the object that you want to
        /// create in the supplied bucket.</param>
        public static async Task WritingAnObjectAsync(IAmazonS3 client, string
bucketName, string keyName)
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    ContentBody = "sample text",
                    ServerSideEncryptionMethod =
ServerSideEncryptionMethod.AES256,
                };

                var putResponse = await client.PutObjectAsync(putRequest);

                // Determine the encryption state of an object.
                GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                };
```

```
                GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);
                ServerSideEncryptionMethod objectEncryption =
response.ServerSideEncryptionMethod;

                Console.WriteLine($"Encryption method used: {0}",
objectEncryption.ToString());
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error: '{ex.Message}' when writing an
object");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for .NET API.

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
```

```
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::putObject(const Aws::String &bucketName,
                           const Aws::String &fileName,
                           const Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client s3Client(clientConfig);

    Aws::S3::Model::PutObjectRequest request;
    request.SetBucket(bucketName);
    //We are using the name of the file as the key for the object in the bucket.
    //However, this is just a string and can be set according to your retrieval
 needs.
    request.SetKey(fileName);

    std::shared_ptr<Aws::IOStream> inputData =
            Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                          fileName.c_str(),
                                          std::ios_base::in |
 std::ios_base::binary);

    if (!*inputData) {
        std::cerr << "Error unable to read file " << fileName << std::endl;
        return false;
    }

    request.SetBody(inputData);

    Aws::S3::Model::PutObjectOutcome outcome =
            s3Client.PutObject(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: putObject: " <<
                  outcome.GetError().GetMessage() << std::endl;
    } else {
```

```
        std::cout << "Added object '" << fileName << "' to bucket '"
                  << bucketName << "'.";
    }

    return outcome.IsSuccess();
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

L'exemple suivant utilise la `put-object` commande pour charger un objet sur Amazon S3 :

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --body
 my_images.tar.bz2
```

L'exemple suivant montre le téléchargement d'un fichier vidéo (le fichier vidéo est spécifié à l'aide de la syntaxe du système de fichiers Windows).) :

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body
 e:\media\videos\f-sharp-3-data-services.mp4
```

Pour plus d'informations sur le téléchargement d'objets, consultez Uploading Objects dans le manuel Amazon S3 Developer Guide.

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Placez un objet dans un compartiment à l'aide de l'API de bas niveau.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// UploadFile reads from a file and puts the data into an object in a bucket.
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,
 fileName string) error {
 file, err := os.Open(fileName)
 if err != nil {
  log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)
 } else {
  defer file.Close()
  _, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
   Bucket: aws.String(bucketName),
   Key:    aws.String(objectKey),
   Body:   file,
  })
  if err != nil {
   log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
    fileName, bucketName, objectKey, err)
  }
 }
 return err
}
```

Téléchargez un objet dans un bucket à l'aide d'un gestionnaire de transferts.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
```

```
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// UploadObject uses the S3 upload manager to upload an object to a bucket.
func (actor S3Actions) UploadObject(ctx context.Context, bucket string, key
 string, contents string) (string, error) {
 var outKey string
 input := &s3.PutObjectInput{
  Bucket:            aws.String(bucket),
  Key:               aws.String(key),
  Body:              bytes.NewReader([]byte(contents)),
  ChecksumAlgorithm: types.ChecksumAlgorithmSha256,
 }
 output, err := actor.S3Manager.Upload(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 } else {
  err := s3.NewObjectExistsWaiter(actor.S3Client).Wait(ctx, &s3.HeadObjectInput{
   Bucket: aws.String(bucket),
   Key:    aws.String(key),
  }, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for object %s to exist in %s.\n", key,
 bucket)
  } else {
   outKey = *output.Key
  }
 }
 return outKey, err
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez un fichier dans un compartiment à l'aide d'un S3Client.

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class PutObject {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <bucketName> <objectKey> <objectPath>\s

                Where:
                  bucketName - The Amazon S3 bucket to upload an object into.
                  objectKey - The object to upload (for example, book.pdf).
                  objectPath - The path where the file is located (for example,
 C:/AWS/book2.pdf).\s
```

```
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        String objectPath = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        putS3Object(s3, bucketName, objectKey, objectPath);
        s3.close();
    }

    // This example uses RequestBody.fromFile to avoid loading the whole file
 into
    // memory.
    public static void putS3Object(S3Client s3, String bucketName, String
 objectKey, String objectPath) {
        try {
            Map<String, String> metadata = new HashMap<>();
            metadata.put("x-amz-meta-myVal", "test");
            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .metadata(metadata)
                    .build();

            s3.putObject(putOb, RequestBody.fromFile(new File(objectPath)));
            System.out.println("Successfully placed " + objectKey + " into bucket
 " + bucketName);

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

Utilisez un S3 TransferManager pour télécharger un fichier dans un compartiment. Consultez le fichier complet et le test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedFileUpload;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;
import software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public String uploadFile(S3TransferManager transferManager, String
 bucketName,
                             String key, URI filePathURI) {
        UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
            .putObjectRequest(b -> b.bucket(bucketName).key(key))
            .source(Paths.get(filePathURI))
            .build();

        FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);

        CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
        return uploadResult.response().eTag();
    }
```

Chargez un objet dans un compartiment et définissez des étiquettes à l'aide d'un S3Client.

```
    public static void putS3ObjectTags(S3Client s3, String bucketName, String
 objectKey, String objectPath) {
        try {
            Tag tag1 = Tag.builder()
                    .key("Tag 1")
                    .value("This is tag 1")
                    .build();
```

```
            Tag tag2 = Tag.builder()
                    .key("Tag 2")
                    .value("This is tag 2")
                    .build();

            List<Tag> tags = new ArrayList<>();
            tags.add(tag1);
            tags.add(tag2);

            Tagging allTags = Tagging.builder()
                    .tagSet(tags)
                    .build();

            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .tagging(allTags)
                    .build();

            s3.putObject(putOb,
RequestBody.fromBytes(getObjectFile(objectPath)));

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void updateObjectTags(S3Client s3, String bucketName, String
objectKey) {
        try {
            GetObjectTaggingRequest taggingRequest =
GetObjectTaggingRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();

            GetObjectTaggingResponse getTaggingRes =
s3.getObjectTagging(taggingRequest);
            List<Tag> obTags = getTaggingRes.tagSet();
            for (Tag sinTag : obTags) {
                System.out.println("The tag key is: " + sinTag.key());
                System.out.println("The tag value is: " + sinTag.value());
            }
```

```
            // Replace the object's tags with two new tags.
            Tag tag3 = Tag.builder()
                    .key("Tag 3")
                    .value("This is tag 3")
                    .build();

            Tag tag4 = Tag.builder()
                    .key("Tag 4")
                    .value("This is tag 4")
                    .build();

            List<Tag> tags = new ArrayList<>();
            tags.add(tag3);
            tags.add(tag4);

            Tagging updatedTags = Tagging.builder()
                    .tagSet(tags)
                    .build();

            PutObjectTaggingRequest taggingRequest1 =
PutObjectTaggingRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .tagging(updatedTags)
                    .build();

            s3.putObjectTagging(taggingRequest1);
            GetObjectTaggingResponse getTaggingRes2 =
s3.getObjectTagging(taggingRequest);
            List<Tag> modTags = getTaggingRes2.tagSet();
            for (Tag sinTag : modTags) {
                System.out.println("The tag key is: " + sinTag.key());
                System.out.println("The tag value is: " + sinTag.value());
            }

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Return a byte array.
    private static byte[] getObjectFile(String filePath) {
```

```
        FileInputStream fileInputStream = null;
        byte[] bytesArray = null;

        try {
            File file = new File(filePath);
            bytesArray = new byte[(int) file.length()];
            fileInputStream = new FileInputStream(file);
            fileInputStream.read(bytesArray);

        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }

        return bytesArray;
    }
}
```

Chargez un objet dans un compartiment et définissez les métadonnées à l'aide d'un S3Client.

```
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
```

```
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class PutObjectMetadata {
    public static void main(String[] args) {
        final String USAGE = """

                Usage:
                  <bucketName> <objectKey> <objectPath>\s

                Where:
                  bucketName - The Amazon S3 bucket to upload an object into.
                  objectKey - The object to upload (for example, book.pdf).
                  objectPath - The path where the file is located (for example,
 C:/AWS/book2.pdf).\s
                """;

        if (args.length != 3) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String bucketName = args[0];
        String objectKey = args[1];
        String objectPath = args[2];
        System.out.println("Putting object " + objectKey + " into bucket " +
 bucketName);
        System.out.println("  in bucket: " + bucketName);
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        putS3Object(s3, bucketName, objectKey, objectPath);
        s3.close();
    }

    // This example uses RequestBody.fromFile to avoid loading the whole file
 into
    // memory.
    public static void putS3Object(S3Client s3, String bucketName, String
 objectKey, String objectPath) {
        try {
            Map<String, String> metadata = new HashMap<>();
```

```
            metadata.put("author", "Mary Doe");
            metadata.put("version", "1.0.0.0");

            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .metadata(metadata)
                    .build();

            s3.putObject(putOb, RequestBody.fromFile(new File(objectPath)));
            System.out.println("Successfully placed " + objectKey + " into bucket
 " + bucketName);

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

Chargez un objet dans un compartiment et définissez une valeur de conservation de l'objet à l'aide d'un S3Client.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.time.Instant;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.ZoneOffset;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class PutObjectRetention {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <key> <bucketName>\s

                Where:
                    key - The name of the object (for example, book.pdf).\s
                    bucketName - The Amazon S3 bucket name that contains the
 object (for example, bucket1).\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String key = args[0];
        String bucketName = args[1];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        setRentionPeriod(s3, key, bucketName);
        s3.close();
    }

    public static void setRentionPeriod(S3Client s3, String key, String bucket) {
        try {
            LocalDate localDate = LocalDate.parse("2020-07-17");
            LocalDateTime localDateTime = localDate.atStartOfDay();
            Instant instant = localDateTime.toInstant(ZoneOffset.UTC);

            ObjectLockRetention lockRetention = ObjectLockRetention.builder()
                    .mode("COMPLIANCE")
                    .retainUntilDate(instant)
                    .build();

            PutObjectRetentionRequest retentionRequest =
 PutObjectRetentionRequest.builder()
                    .bucket(bucket)
```

```
                    .key(key)
                    .bypassGovernanceRetention(true)
                    .retention(lockRetention)
                    .build();

            // To set Retention on an object, the Amazon S3 bucket must support
 object
            // locking, otherwise an exception is thrown.
            s3.putObjectRetention(retentionRequest);
            System.out.print("An object retention configuration was successfully
 placed on the object");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez l'objet.

```
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";

const client = new S3Client({});

export const main = async () => {
  const command = new PutObjectCommand({
    Bucket: "test-bucket",
```

```
    Key: "hello-s3.txt",
    Body: "Hello S3!",
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun putS3Object(
    bucketName: String,
    objectKey: String,
    objectPath: String,
) {
    val metadataVal = mutableMapOf<String, String>()
    metadataVal["myVal"] = "test"

    val request =
        PutObjectRequest {
            bucket = bucketName
            key = objectKey
            metadata = metadataVal
```

```
            body = File(objectPath).asByteStream()
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.putObject(request)
        println("Tag information is ${response.eTag}")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section AWS SDK pour la référence de l'API Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez un objet dans un compartiment.

```
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);

        $fileName = __DIR__ . "/local-file-" . uniqid();
        try {
            $this->s3client->putObject([
                'Bucket' => $this->bucketName,
                'Key' => $fileName,
                'SourceFile' => __DIR__ . '/testfile.txt'
            ]);
            echo "Uploaded $fileName to $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to upload $fileName with error: " . $exception-
>getMessage();
            exit("Please fix error with file upload before continuing.");
        }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for PHP API.

PowerShell

Outils pour PowerShell

Exemple 1 : cette commande télécharge le fichier unique « local-sample.txt » sur Amazon S3, créant un objet avec la clé « sample.txt » dans le compartiment « test-files ».

```
Write-S3Object -BucketName test-files -Key "sample.txt" -File .\local-sample.txt
```

Exemple 2 : cette commande télécharge le fichier unique « sample.txt » sur Amazon S3, créant un objet avec la clé « sample.txt » dans le compartiment « test-files ». Si le paramètre -Key n'est pas fourni, le nom du fichier est utilisé comme clé d'objet S3.

```
Write-S3Object -BucketName test-files -File .\sample.txt
```

Exemple 3 : cette commande télécharge le fichier unique « local-sample.txt » sur Amazon S3, créant un objet avec la clé « prefix/to/sample.txt » dans le compartiment « test-files ».

```
Write-S3Object -BucketName test-files -Key "prefix/to/sample.txt" -File .\local-
sample.txt
```

Exemple 4 : Cette commande télécharge tous les fichiers du sous-répertoire « Scripts » vers le bucket « test-files » et applique le préfixe de clé commun « » à chaque objet. SampleScripts Chaque fichier téléchargé aura une clé « SampleScripts /filename » où « filename » varie.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
```

Exemple 5 : Cette commande télécharge tous les fichiers*.ps1 du répertoire local « Scripts » vers le bucket « test-files » et applique le préfixe de clé commun « » à chaque objet. SampleScripts Chaque fichier téléchargé aura une clé « SampleScripts /filename.ps1 » où le « nom de fichier » varie.

```
Write-S3Object -BucketName test-files -Folder .\Scripts -KeyPrefix SampleScripts\
 -SearchPattern *.ps1
```

Exemple 6 : Cette commande crée un nouvel objet S3 contenant la chaîne de contenu spécifiée avec la clé « sample.txt ».

```
Write-S3Object -BucketName test-files -Key "sample.txt" -Content "object
 contents"
```

Exemple 7 : Cette commande télécharge le fichier spécifié (le nom du fichier est utilisé comme clé) et applique les balises spécifiées au nouvel objet.

```
Write-S3Object -BucketName test-files -File "sample.txt" -TagSet
 @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

Exemple 8 : Cette commande télécharge de manière récursive le dossier spécifié et applique les balises spécifiées à tous les nouveaux objets.

```
Write-S3Object -BucketName test-files -Folder . -KeyPrefix "TaggedFiles" -Recurse
 -TagSet @{Key="key1";Value="value1"},@{Key="key2";Value="value2"}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource
 in Boto3
                          that wraps object actions in a class-like structure.
        """
```

```
        self.object = s3_object
        self.key = self.object.key


    def put(self, data):
        """
        Upload data to the object.

        :param data: The data to upload. This can either be bytes or a string.
When this
                     argument is a string, it is interpreted as a file name,
which is
                     opened in read bytes mode.
        """
        put_data = data
        if isinstance(data, str):
            try:
                put_data = open(data, "rb")
            except IOError:
                logger.exception("Expected file name or binary data, got '%s'.",
data)
                raise

        try:
            self.object.put(Body=put_data)
            self.object.wait_until_exists()
            logger.info(
                "Put object '%s' to bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't put object '%s' to bucket '%s'.",
                self.object.key,
                self.object.bucket_name,
            )
            raise
        finally:
            if getattr(put_data, "close", None):
                put_data.close()
```

- Pour plus de détails sur l'API, consultez PutObjectle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez un fichier à l'aide d'un chargeur géré (Object.upload_file).

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
```

```
  bucket_name = "doc-example-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"

  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Chargez un fichier en utilisant la commande Object.put.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, "rb") do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-object-key"
```

```
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Chargez un fichier en utilisant la commande Object.put et ajoutez le chiffrement côté serveur.

```
require "aws-sdk-s3"

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
 object_content))
```

```
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section Référence des AWS SDK for Ruby API.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
pub async fn upload_object(
    client: &Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<PutObjectOutput, SdkError<PutObjectError>> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
}
```

- Pour plus de détails sur l'API, voir PutObjectla section de référence de l'API AWS SDK for Rust.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
"Get contents of file from application server."
DATA lv_body TYPE xstring.
OPEN DATASET iv_file_name FOR INPUT IN BINARY MODE.
READ DATASET iv_file_name INTO lv_body.
CLOSE DATASET iv_file_name.

"Upload/put an object to an S3 bucket."
TRY.
    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_file_name
        iv_body = lv_body
    ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section de référence du AWS SDK pour l'API SAP ABAP.

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Téléchargez un fichier du stockage local vers un compartiment.

```
    public func uploadFile(bucket: String, key: String, file: String) async
throws {
        let fileUrl = URL(fileURLWithPath: file)
        let fileData = try Data(contentsOf: fileUrl)
        let dataStream = ByteStream.from(data: fileData)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

Chargez le contenu d'un objet de données Swift vers un compartiment.

```
    public func createFile(bucket: String, key: String, withData data: Data)
async throws {
        let dataStream = ByteStream.from(data: data)

        let input = PutObjectInput(
            body: dataStream,
```

```
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectà la section AWS SDK pour la référence de l'API Swift.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutObjectAcl`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutObjectAcl`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Gérer les listes de contrôle d'accès (ACL)

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
bool AwsDoc::S3::putObjectAcl(const Aws::String &bucketName, const Aws::String
 &objectKey, const Aws::String &ownerID,
                              const Aws::String &granteePermission, const
 Aws::String &granteeType,
                              const Aws::String &granteeID, const Aws::String
 &granteeEmailAddress,
```

```
                                const Aws::String &granteeURI, const
Aws::S3::S3ClientConfiguration &clientConfig) {
   Aws::S3::S3Client s3Client(clientConfig);

   Aws::S3::Model::Owner owner;
   owner.SetID(ownerID);

   Aws::S3::Model::Grantee grantee;
   grantee.SetType(setGranteeType(granteeType));

   if (!granteeEmailAddress.empty()) {
       grantee.SetEmailAddress(granteeEmailAddress);
   }

   if (!granteeID.empty()) {
       grantee.SetID(granteeID);
   }

   if (!granteeURI.empty()) {
       grantee.SetURI(granteeURI);
   }

   Aws::S3::Model::Grant grant;
   grant.SetGrantee(grantee);
   grant.SetPermission(setGranteePermission(granteePermission));

   Aws::Vector<Aws::S3::Model::Grant> grants;
   grants.push_back(grant);

   Aws::S3::Model::AccessControlPolicy acp;
   acp.SetOwner(owner);
   acp.SetGrants(grants);

   Aws::S3::Model::PutObjectAclRequest request;
   request.SetAccessControlPolicy(acp);
   request.SetBucket(bucketName);
   request.SetKey(objectKey);

   Aws::S3::Model::PutObjectAclOutcome outcome =
           s3Client.PutObjectAcl(request);

   if (!outcome.IsSuccess()) {
       auto error = outcome.GetError();
       std::cerr << "Error: putObjectAcl: " << error.GetExceptionName()
```

```
                  << " - " << error.GetMessage() << std::endl;
    } else {
        std::cout << "Successfully added an ACL to the object '" << objectKey
                  << "' in the bucket '" << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which converts a human-readable string to a built-in type
 enumeration.
/*!
 \param access: Human readable string.
 \return Permission: Permission enumeration.
*/
Aws::S3::Model::Permission setGranteePermission(const Aws::String &access) {
    if (access == "FULL_CONTROL")
        return Aws::S3::Model::Permission::FULL_CONTROL;
    if (access == "WRITE")
        return Aws::S3::Model::Permission::WRITE;
    if (access == "READ")
        return Aws::S3::Model::Permission::READ;
    if (access == "WRITE_ACP")
        return Aws::S3::Model::Permission::WRITE_ACP;
    if (access == "READ_ACP")
        return Aws::S3::Model::Permission::READ_ACP;
    return Aws::S3::Model::Permission::NOT_SET;
}

//! Routine which converts a human-readable string to a built-in type
 enumeration.
/*!
 \param type: Human readable string.
 \return Type: Type enumeration.
*/
Aws::S3::Model::Type setGranteeType(const Aws::String &type) {
    if (type == "Amazon customer by email")
        return Aws::S3::Model::Type::AmazonCustomerByEmail;
    if (type == "Canonical user")
        return Aws::S3::Model::Type::CanonicalUser;
    if (type == "Group")
        return Aws::S3::Model::Type::Group;
    return Aws::S3::Model::Type::NOT_SET;
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectAclà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

La commande suivante accorde l'`read`autorisation `full control` à deux AWS utilisateurs (user1@example.com et user2@example.com) et à tout le monde :

```
aws s3api put-object-acl --bucket MyBucket --key file.txt --grant-full-control
 emailaddress=user1@example.com,emailaddress=user2@example.com --grant-read
 uri=http://acs.amazonaws.com/groups/global/AllUsers
```

Consultez http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html pour plus de détails sur les ACL personnalisées (les commandes ACL s3api, par exemple`put-object-acl`, utilisent la même notation d'argument abrégée).

- Pour plus de détails sur l'API, reportez-vous PutObjectAclà la section Référence des AWS CLI commandes.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
```

```
        :param s3_object: A Boto3 Object resource. This is a high-level resource
in Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key


    def put_acl(self, email):
        """
        Applies an ACL to the object that grants read access to an AWS user
identified
        by email address.

        :param email: The email address of the user to grant access.
        """
        try:
            acl = self.object.Acl()
            # Putting an ACL overwrites the existing ACL, so append new grants
            # if you want to preserve existing grants.
            grants = acl.grants if acl.grants else []
            grants.append(
                {
                    "Grantee": {"Type": "AmazonCustomerByEmail", "EmailAddress":
email},
                    "Permission": "READ",
                }
            )
            acl.put(AccessControlPolicy={"Grants": grants, "Owner": acl.owner})
            logger.info("Granted read access to %s.", email)
        except ClientError:
            logger.exception("Couldn't add ACL to object '%s'.", self.object.key)
            raise
```

- Pour plus de détails sur l'API, consultez PutObjectAcllle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **PutObjectLegalHold** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutObjectLegalHold`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/// <summary>
/// Set or modify a legal hold on an object in an S3 bucket.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The key of the object.</param>
/// <param name="holdStatus">The On or Off status for the legal hold.</param>
/// <returns>True if successful.</returns>
public async Task<bool> ModifyObjectLegalHold(string bucketName,
    string objectKey, ObjectLockLegalHoldStatus holdStatus)
{
    try
    {
        var request = new PutObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey,
            LegalHold = new ObjectLockLegalHold()
            {
                Status = holdStatus
            }
        };
```

```
            var response = await _amazonS3.PutObjectLegalHoldAsync(request);
            Console.WriteLine($"\tModified legal hold for {objectKey} in
 {bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tError modifying legal hold: '{ex.Message}'");
            return false;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour appliquer une suspension légale à un objet

L'`put-object-legal-hold`exemple suivant définit une conservation légale sur l'objet`doc1.rtf`.

```
aws s3api put-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --legal-hold Status=ON
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// PutObjectLegalHold sets the legal hold configuration for an S3 object.
func (actor S3Actions) PutObjectLegalHold(ctx context.Context, bucket string, key
 string, versionId string, legalHoldStatus types.ObjectLockLegalHoldStatus) error
 {
 input := &s3.PutObjectLegalHoldInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
  LegalHold: &types.ObjectLockLegalHold{
   Status: legalHoldStatus,
  },
 }
 if versionId != "" {
  input.VersionId = aws.String(versionId)
 }

 _, err := actor.S3Client.PutObjectLegalHold(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  }
 }
```

```
 return err
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   // Set or modify a legal hold on an object in an S3 bucket.
   public void modifyObjectLegalHold(String bucketName, String objectKey,
boolean legalHoldOn) {
       ObjectLockLegalHold legalHold ;
       if (legalHoldOn) {
           legalHold = ObjectLockLegalHold.builder()
               .status(ObjectLockLegalHoldStatus.ON)
               .build();
       } else {
           legalHold = ObjectLockLegalHold.builder()
               .status(ObjectLockLegalHoldStatus.OFF)
               .build();
       }

       PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
           .bucket(bucketName)
           .key(objectKey)
           .legalHold(legalHold)
           .build();

       getClient().putObjectLegalHold(legalHoldRequest) ;
```

```
        System.out.println("Modified legal hold for "+ objectKey +" in
 "+bucketName +".");
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import { PutObjectLegalHoldCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new PutObjectLegalHoldCommand({
    Bucket: bucketName,
    Key: objectKey,
    LegalHold: {
      // Set the status to 'ON' to place a legal hold on the object.
      // Set the status to 'OFF' to remove the legal hold.
      Status: "ON",
    },
    // Optionally, you can provide additional parameters
    // ChecksumAlgorithm: "ALGORITHM",
    // ContentMD5: "MD5_HASH",
    // ExpectedBucketOwner: "ACCOUNT_ID",
```

```
    // RequestPayer: "requester",
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const response = await client.send(command);
    console.log(
      `Object legal hold status: ${response.$metadata.httpStatusCode}`,
    );
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLegalHoldà la section Référence des AWS SDK for JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutObjectLockConfiguration`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutObjectLockConfiguration`.

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l’exemple de code suivant :

- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez la configuration du verrouillage des objets d'un bucket.

```
    /// <summary>
    /// Enable object lock on an existing bucket.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to modify.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableObjectLockOnBucket(string bucketName)
    {
        try
        {
            // First, enable Versioning on the bucket.
            await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig()
                {
                    EnableMfaDelete = false,
                    Status = VersionStatus.Enabled
                }
            });

            var request = new PutObjectLockConfigurationRequest()
            {
                BucketName = bucketName,
                ObjectLockConfiguration = new ObjectLockConfiguration()
                {
                    ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
                },
            };
```

```
            var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
            Console.WriteLine($"\tAdded an object lock policy to bucket
{bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
            return false;
        }
    }
```

Définissez la période de rétention par défaut d'un bucket.

```
    /// <summary>
    /// Set or modify a retention period on an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket to modify.</param>
    /// <param name="retention">The retention mode.</param>
    /// <param name="retainUntilDate">The date for retention until.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)
    {
        var enabledString = enableObjectLock ? "Enabled" : "Disabled";
        var timeDifference = retainUntilDate.Subtract(DateTime.Now);
        try
        {
            // First, enable Versioning on the bucket.
            await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig()
                {
                    EnableMfaDelete = false,
                    Status = VersionStatus.Enabled
                }
            });

            var request = new PutObjectLockConfigurationRequest()
```

```
            {
                BucketName = bucketName,
                ObjectLockConfiguration = new ObjectLockConfiguration()
                {
                    ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                    Rule = new ObjectLockRule()
                    {
                        DefaultRetention = new DefaultRetention()
                        {
                            Mode = retention,
                            Days = timeDifference.Days // Can be specified in
days or years but not both.
                        }
                    }
                }
            };

            var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
            Console.WriteLine($"\tAdded a default retention to bucket
{bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tError modifying object lock: '{ex.Message}'");
            return false;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour définir une configuration de verrouillage d'objets sur un bucket

L'`put-object-lock-configuration`exemple suivant définit un verrouillage d'objet de 50 jours sur le compartiment spécifié.

```
aws s3api put-object-lock-configuration \
    --bucket my-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez la configuration du verrouillage des objets d'un bucket.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// EnableObjectLockOnBucket enables object locking on an existing bucket.
func (actor S3Actions) EnableObjectLockOnBucket(ctx context.Context, bucket
 string) error {
 // Versioning must be enabled on the bucket before object locking is enabled.
 verInput := &s3.PutBucketVersioningInput{
  Bucket: aws.String(bucket),
  VersioningConfiguration: &types.VersioningConfiguration{
   MFADelete: types.MFADeleteDisabled,
   Status:    types.BucketVersioningStatusEnabled,
  },
 }
```

```
 _, err := actor.S3Client.PutBucketVersioning(ctx, verInput)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
  return err
 }

 input := &s3.PutObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
  ObjectLockConfiguration: &types.ObjectLockConfiguration{
   ObjectLockEnabled: types.ObjectLockEnabledEnabled,
  },
 }
 _, err = actor.S3Client.PutObjectLockConfiguration(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 }

 return err
}
```

Définissez la période de rétention par défaut d'un bucket.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// ModifyDefaultBucketRetention modifies the default retention period of an
 existing bucket.
```

```
func (actor S3Actions) ModifyDefaultBucketRetention(
 ctx context.Context, bucket string, lockMode types.ObjectLockEnabled,
 retentionPeriod int32, retentionMode types.ObjectLockRetentionMode) error {

 input := &s3.PutObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
  ObjectLockConfiguration: &types.ObjectLockConfiguration{
   ObjectLockEnabled: lockMode,
   Rule: &types.ObjectLockRule{
    DefaultRetention: &types.DefaultRetention{
     Days: aws.Int32(retentionPeriod),
     Mode: retentionMode,
    },
   },
  },
 }
 _, err := actor.S3Client.PutObjectLockConfiguration(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 }

 return err
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez la configuration du verrouillage des objets d'un bucket.

```
   // Enable object lock on an existing bucket.
   public void enableObjectLockOnBucket(String bucketName) {
       try {
           VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
               .status(BucketVersioningStatus.ENABLED)
               .build();

           PutBucketVersioningRequest putBucketVersioningRequest =
PutBucketVersioningRequest.builder()
               .bucket(bucketName)
               .versioningConfiguration(versioningConfiguration)
               .build();

           // Enable versioning on the bucket.
           getClient().putBucketVersioning(putBucketVersioningRequest);
           PutObjectLockConfigurationRequest request =
PutObjectLockConfigurationRequest.builder()
               .bucket(bucketName)
               .objectLockConfiguration(ObjectLockConfiguration.builder()
                   .objectLockEnabled(ObjectLockEnabled.ENABLED)
                   .build())
               .build();

           getClient().putObjectLockConfiguration(request);
           System.out.println("Successfully enabled object lock on
"+bucketName);

       } catch (S3Exception ex) {
           System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
       }
   }
```

Définissez la période de rétention par défaut d'un bucket.

```
   // Set or modify a retention period on an S3 bucket.
   public void modifyBucketDefaultRetention(String bucketName) {
       VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
```

```
            .mfaDelete(MFADelete.DISABLED)
            .status(BucketVersioningStatus.ENABLED)
            .build();

        PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
            .bucket(bucketName)
            .versioningConfiguration(versioningConfiguration)
            .build();

        getClient().putBucketVersioning(versioningRequest);
        DefaultRetention rention = DefaultRetention.builder()
            .days(1)
            .mode(ObjectLockRetentionMode.GOVERNANCE)
            .build();

        ObjectLockRule lockRule = ObjectLockRule.builder()
            .defaultRetention(rention)
            .build();

        ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
            .objectLockEnabled(ObjectLockEnabled.ENABLED)
            .rule(lockRule)
            .build();

        PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .objectLockConfiguration(objectLockConfiguration)
            .build();


getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
        System.out.println("Added a default retention to bucket "+bucketName
+".");
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez la configuration du verrouillage des objets d'un bucket.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import {
  PutObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new PutObjectLockConfigurationCommand({
    Bucket: bucketName,
    // The Object Lock configuration that you want to apply to the specified
 bucket.
    ObjectLockConfiguration: {
      ObjectLockEnabled: "Enabled",
    },
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // Token: "OPTIONAL_TOKEN",
  });

  try {
    const response = await client.send(command);
    console.log(
      `Object Lock Configuration updated: ${response.$metadata.httpStatusCode}`,
    );
```

```
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}
```

Définissez la période de rétention par défaut d'un bucket.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import {
  PutObjectLockConfigurationCommand,
  S3Client,
} from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 */
export const main = async (client, bucketName) => {
  const command = new PutObjectLockConfigurationCommand({
    Bucket: bucketName,
    // The Object Lock configuration that you want to apply to the specified
 bucket.
    ObjectLockConfiguration: {
      ObjectLockEnabled: "Enabled",
      Rule: {
        DefaultRetention: {
          Mode: "GOVERNANCE",
          Years: 3,
        },
      },
    },
    // Optionally, you can provide additional parameters
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // Token: "OPTIONAL_TOKEN",
```

```
  });

  try {
    const response = await client.send(command);
    console.log(
      `Default Object Lock Configuration updated: ${response.
$metadata.httpStatusCode}`,
    );
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME");
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectLockConfigurationà la section Référence des AWS SDK for JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutObjectRetention`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`PutObjectRetention`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans l'exemple de code suivant :

- Verrouiller des objets Amazon S3

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    /// <summary>
    /// Set or modify a retention period on an object in an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The key of the object.</param>
    /// <param name="retention">The retention mode.</param>
    /// <param name="retainUntilDate">The date retention expires.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
        string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
    {
        try
        {
            var request = new PutObjectRetentionRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
                Retention = new ObjectLockRetention()
                {
                    Mode = retention,
                    RetainUntilDate = retainUntilDate
                }
            };

            var response = await _amazonS3.PutObjectRetentionAsync(request);
            Console.WriteLine($"\tSet retention for {objectKey} in {bucketName}
until {retainUntilDate:d}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
```

```
            Console.WriteLine($"\tError modifying retention period:
'{ex.Message}'");
            return false;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour définir une configuration de rétention d'objets pour un objet

L'`put-object-retention`exemple suivant définit une configuration de rétention d'objet pour l'objet spécifié jusqu'au 01/01/2025.

```
aws s3api put-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate":
"2025-01-01T00:00:00" }'
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS CLI commandes.

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// PutObjectRetention sets the object retention configuration for an S3 object.
func (actor S3Actions) PutObjectRetention(ctx context.Context, bucket string, key
 string, retentionMode types.ObjectLockRetentionMode, retentionPeriodDays int32)
 error {
 input := &s3.PutObjectRetentionInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
  Retention: &types.ObjectLockRetention{
   Mode:            retentionMode,
   RetainUntilDate: aws.Time(time.Now().AddDate(0, 0, int(retentionPeriodDays))),
  },
  BypassGovernanceRetention: aws.Bool(true),
 }

 _, err := actor.S3Client.PutObjectRetention(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  }
 }

 return err
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS SDK for Go API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   // Set or modify a retention period on an object in an S3 bucket.
   public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
{
       // Calculate the instant one day from now.
       Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

       // Convert the Instant to a ZonedDateTime object with a specific time
zone.
       ZonedDateTime zonedDateTime =
futureInstant.atZone(ZoneId.systemDefault());

       // Define a formatter for human-readable output.
       DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss");

       // Format the ZonedDateTime object to a human-readable date string.
       String humanReadableDate = formatter.format(zonedDateTime);

       // Print the formatted date string.
       System.out.println("Formatted Date: " + humanReadableDate);
       ObjectLockRetention retention = ObjectLockRetention.builder()
           .mode(ObjectLockRetentionMode.GOVERNANCE)
           .retainUntilDate(futureInstant)
           .build();

       PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
           .bucket(bucketName)
           .key(objectKey)
           .retention(retention)
           .build();
```

```
        getClient().putObjectRetention(retentionRequest);
        System.out.println("Set retention for "+objectKey +" in " +bucketName +"
 until "+ humanReadableDate +".");
    }
```

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { fileURLToPath } from "url";
import { PutObjectRetentionCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new PutObjectRetentionCommand({
    Bucket: bucketName,
    Key: objectKey,
    BypassGovernanceRetention: false,
    // ChecksumAlgorithm: "ALGORITHM",
    // ContentMD5: "MD5_HASH",
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    Retention: {
      Mode: "GOVERNANCE", // or "COMPLIANCE"
      RetainUntilDate: new Date(new Date().getTime() + 24 * 60 * 60 * 1000),
```

```
    },
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const response = await client.send(command);
    console.log(
      `Object Retention settings updated: ${response.$metadata.httpStatusCode}`,
    );
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "BUCKET_NAME", "OBJECT_KEY");
}
```

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS SDK for JavaScript API.

PowerShell

Outils pour PowerShell

Exemple 1 : La commande active le mode de rétention de la gouvernance jusqu'à la date du 31 décembre 2019 00:00:00 pour l'objet « testfile.txt » dans le compartiment S3 donné.

```
Write-S3ObjectRetention -BucketName 's3buckettesting' -Key 'testfile.txt' -
Retention_Mode GOVERNANCE -Retention_RetainUntilDate "2019-12-31T00:00:00"
```

- Pour plus de détails sur l'API, reportez-vous PutObjectRetentionà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`RestoreObject`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`RestoreObject`.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to restore an archived object in an Amazon
/// Simple Storage Service (Amazon S3) bucket.
/// </summary>
public class RestoreArchivedObject
{
    public static void Main()
    {
        string bucketName = "doc-example-bucket";
        string objectKey = "archived-object.txt";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        IAmazonS3 client = new AmazonS3Client(bucketRegion);
        RestoreObjectAsync(client, bucketName, objectKey).Wait();
    }

    /// <summary>
    /// This method restores an archived object from an Amazon S3 bucket.
    /// </summary>
```

```
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// RestoreObjectAsync.</param>
        /// <param name="bucketName">A string representing the name of the
        /// bucket where the object was located before it was archived.</param>
        /// <param name="objectKey">A string representing the name of the
        /// archived object to restore.</param>
        public static async Task RestoreObjectAsync(IAmazonS3 client, string
bucketName, string objectKey)
        {
            try
            {
                var restoreRequest = new RestoreObjectRequest
                {
                    BucketName = bucketName,
                    Key = objectKey,
                    Days = 2,
                };
                RestoreObjectResponse response = await
client.RestoreObjectAsync(restoreRequest);

                // Check the status of the restoration.
                await CheckRestorationStatusAsync(client, bucketName, objectKey);
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine($"Error: {amazonS3Exception.Message}");
            }
        }

        /// <summary>
        /// This method retrieves the status of the object's restoration.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// GetObjectMetadataAsync.</param>
        /// <param name="bucketName">A string representing the name of the Amazon
        /// S3 bucket which contains the archived object.</param>
        /// <param name="objectKey">A string representing the name of the
        /// archived object you want to restore.</param>
        public static async Task CheckRestorationStatusAsync(IAmazonS3 client,
string bucketName, string objectKey)
        {
```

```
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
            };

            GetObjectMetadataResponse response = await
client.GetObjectMetadataAsync(metadataRequest);

            var restStatus = response.RestoreInProgress ? "in-progress" :
"finished or failed";
            Console.WriteLine($"Restoration status: {restStatus}");
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous RestoreObjectà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour créer une demande de restauration pour un objet

L'`restore-object`exemple suivant restaure l'objet Amazon S3 Glacier spécifié pour le compartiment `my-glacier-bucket` pendant 10 jours.

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
    --key doc1.rtf \
    --restore-request Days=10
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous RestoreObjectà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.RestoreRequest;
import software.amazon.awssdk.services.s3.model.GlacierJobParameters;
import software.amazon.awssdk.services.s3.model.RestoreObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.Tier;

/*
 *  For more information about restoring an object, see "Restoring an archived
 object" at
 *  https://docs.aws.amazon.com/AmazonS3/latest/userguide/restoring-objects.html
 *
 *  Before running this Java V2 code example, set up your development
 environment, including your credentials.
 *
 *  For more information, see the following documentation topic:
 *
 *  https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class RestoreObject {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <bucketName> <keyName> <expectedBucketOwner>

                Where:
                    bucketName - The Amazon S3 bucket name.\s
                    keyName - The key name of an object with a Storage class
 value of Glacier.\s
```

```
                    expectedBucketOwner - The account that owns the bucket (you
can obtain this value from the AWS Management Console).\s
                """";

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String keyName = args[1];
        String expectedBucketOwner = args[2];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        restoreS3Object(s3, bucketName, keyName, expectedBucketOwner);
        s3.close();
    }

    public static void restoreS3Object(S3Client s3, String bucketName, String
keyName, String expectedBucketOwner) {
        try {
            RestoreRequest restoreRequest = RestoreRequest.builder()
                    .days(10)

.glacierJobParameters(GlacierJobParameters.builder().tier(Tier.STANDARD).build())
                    .build();

            RestoreObjectRequest objectRequest = RestoreObjectRequest.builder()
                    .expectedBucketOwner(expectedBucketOwner)
                    .bucket(bucketName)
                    .key(keyName)
                    .restoreRequest(restoreRequest)
                    .build();

            s3.restoreObject(objectRequest);

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous RestoreObjectà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`SelectObjectContent`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`SelectObjectContent`.

CLI

AWS CLI

Pour filtrer le contenu d'un objet Amazon S3 en fonction d'une instruction SQL

L'`select-object-content`exemple suivant filtre l'objet `my-data-file.csv` avec l'instruction SQL spécifiée et envoie la sortie dans un fichier.

```
aws s3api select-object-content \
    --bucket my-bucket \
    --key my-data-file.csv \
    --expression "select * from s3object limit 100" \
    --expression-type 'SQL' \
    --input-serialization '{"CSV": {}, "CompressionType": "NONE"}' \
    --output-serialization '{"CSV": {}}' "output.csv"
```

Cette commande ne produit aucun résultat.

- Pour plus de détails sur l'API, reportez-vous SelectObjectContentà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

L'exemple suivant montre une requête utilisant un objet JSON. L'exemple complet montre également l'utilisation d'un objet CSV.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.CSVInput;
import software.amazon.awssdk.services.s3.model.CSVOutput;
import software.amazon.awssdk.services.s3.model.CompressionType;
import software.amazon.awssdk.services.s3.model.ExpressionType;
import software.amazon.awssdk.services.s3.model.FileHeaderInfo;
import software.amazon.awssdk.services.s3.model.InputSerialization;
import software.amazon.awssdk.services.s3.model.JSONInput;
import software.amazon.awssdk.services.s3.model.JSONOutput;
import software.amazon.awssdk.services.s3.model.JSONType;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.services.s3.model.OutputSerialization;
import software.amazon.awssdk.services.s3.model.Progress;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.SelectObjectContentRequest;
import
 software.amazon.awssdk.services.s3.model.SelectObjectContentResponseHandler;
import software.amazon.awssdk.services.s3.model.Stats;

import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;
```

```
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

public class SelectObjectContentExample {
    static final Logger logger =
 LoggerFactory.getLogger(SelectObjectContentExample.class);
    static final String BUCKET_NAME = "select-object-content-" +
 UUID.randomUUID();
    static final S3AsyncClient s3AsyncClient = S3AsyncClient.create();
    static String FILE_CSV = "csv";
    static String FILE_JSON = "json";
    static String URL_CSV = "https://raw.githubusercontent.com/mledoze/countries/
master/dist/countries.csv";
    static String URL_JSON = "https://raw.githubusercontent.com/mledoze/
countries/master/dist/countries.json";

    public static void main(String[] args) {
        SelectObjectContentExample selectObjectContentExample = new
 SelectObjectContentExample();
        try {
            SelectObjectContentExample.setUp();
            selectObjectContentExample.runSelectObjectContentMethodForJSON();
            selectObjectContentExample.runSelectObjectContentMethodForCSV();
        } catch (SdkException e) {
            logger.error(e.getMessage(), e);
            System.exit(1);
        } finally {
            SelectObjectContentExample.tearDown();
        }
    }

    EventStreamInfo runSelectObjectContentMethodForJSON() {
        // Set up request parameters.
        final String queryExpression = "select * from s3object[*][*] c where
 c.area < 350000";
        final String fileType = FILE_JSON;

        InputSerialization inputSerialization = InputSerialization.builder()
                .json(JSONInput.builder().type(JSONType.DOCUMENT).build())
                .compressionType(CompressionType.NONE)
                .build();

        OutputSerialization outputSerialization = OutputSerialization.builder()
                .json(JSONOutput.builder().recordDelimiter(null).build())
```

```
                .build();

        // Build the SelectObjectContentRequest.
        SelectObjectContentRequest select = SelectObjectContentRequest.builder()
                .bucket(BUCKET_NAME)
                .key(FILE_JSON)
                .expression(queryExpression)
                .expressionType(ExpressionType.SQL)
                .inputSerialization(inputSerialization)
                .outputSerialization(outputSerialization)
                .build();

        EventStreamInfo eventStreamInfo = new EventStreamInfo();
        // Call the selectObjectContent method with the request and a response
handler.
        // Supply an EventStreamInfo object to the response handler to gather
records and information from the response.
        s3AsyncClient.selectObjectContent(select,
buildResponseHandler(eventStreamInfo)).join();

        // Log out information gathered while processing the response stream.
        long recordCount = eventStreamInfo.getRecords().stream().mapToInt(record
->
                record.split("\n").length
        ).sum();
        logger.info("Total records {}: {}", fileType, recordCount);
        logger.info("Visitor onRecords for fileType {} called {} times",
fileType, eventStreamInfo.getCountOnRecordsCalled());
        logger.info("Visitor onStats for fileType {}, {}", fileType,
eventStreamInfo.getStats());
        logger.info("Visitor onContinuations for fileType {}, {}", fileType,
eventStreamInfo.getCountContinuationEvents());
        return eventStreamInfo;
    }

    static SelectObjectContentResponseHandler
buildResponseHandler(EventStreamInfo eventStreamInfo) {
        // Use a Visitor to process the response stream. This visitor logs
information and gathers details while processing.
        final SelectObjectContentResponseHandler.Visitor visitor =
SelectObjectContentResponseHandler.Visitor.builder()
                .onRecords(r -> {
                    logger.info("Record event received.");
                    eventStreamInfo.addRecord(r.payload().asUtf8String());
```

```
                    eventStreamInfo.incrementOnRecordsCalled();
                })
                .onCont(ce -> {
                    logger.info("Continuation event received.");
                    eventStreamInfo.incrementContinuationEvents();
                })
                .onProgress(pe -> {
                    Progress progress = pe.details();
                    logger.info("Progress event received:\n bytesScanned:
{}\nbytesProcessed: {}\nbytesReturned:{}",
                            progress.bytesScanned(),
                            progress.bytesProcessed(),
                            progress.bytesReturned());
                })
                .onEnd(ee -> logger.info("End event received."))
                .onStats(se -> {
                    logger.info("Stats event received.");
                    eventStreamInfo.addStats(se.details());
                })
                .build();

        // Build the SelectObjectContentResponseHandler with the visitor that
 processes the stream.
        return SelectObjectContentResponseHandler.builder()
                .subscriber(visitor).build();
    }

    // The EventStreamInfo class is used to store information gathered while
 processing the response stream.
    static class EventStreamInfo {
        private final List<String> records = new ArrayList<>();
        private Integer countOnRecordsCalled = 0;
        private Integer countContinuationEvents = 0;
        private Stats stats;

        void incrementOnRecordsCalled() {
            countOnRecordsCalled++;
        }

        void incrementContinuationEvents() {
            countContinuationEvents++;
        }

        void addRecord(String record) {
```

```
            records.add(record);
        }

        void addStats(Stats stats) {
            this.stats = stats;
        }

        public List<String> getRecords() {
            return records;
        }

        public Integer getCountOnRecordsCalled() {
            return countOnRecordsCalled;
        }

        public Integer getCountContinuationEvents() {
            return countContinuationEvents;
        }

        public Stats getStats() {
            return stats;
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous SelectObjectContentà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`UploadPart`** avec un AWS SDK ou une CLI

Les exemples de code suivants montrent comment utiliser`UploadPart`.

Les exemples d'actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Réalisation d'un chargement partitionné

- Utiliser les totaux de contrôle

CLI

AWS CLI

La commande suivante télécharge la première partie d'un téléchargement en plusieurs parties initié par la `create-multipart-upload` commande :

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --
body part01 --upload-id
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3
```

L'`body`option prend le nom ou le chemin d'un fichier local à télécharger (n'utilisez pas le préfixe file ://). La taille minimale de la pièce est de 5 Mo. L'identifiant de téléchargement est renvoyé par `create-multipart-upload` et peut également être récupéré avec`list-multipart-uploads`. Le bucket et la clé sont spécifiés lorsque vous créez le téléchargement partitionné.

Sortie :

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

Enregistrez la valeur ETag de chaque pièce pour plus tard. Ils sont nécessaires pour effectuer le téléchargement en plusieurs parties.

- Pour plus de détails sur l'API, reportez-vous UploadPartà la section Référence des AWS CLI commandes.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        let upload_part_res = client
            .upload_part()
            .key(&key)
            .bucket(&bucket_name)
            .upload_id(upload_id)
            .body(stream)
            .part_number(part_number)
            .send()
            .await?;
        upload_parts.push(
            CompletedPart::builder()
                .e_tag(upload_part_res.e_tag.unwrap_or_default())
                .part_number(part_number)
                .build(),
        );

   let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
        .set_parts(Some(upload_parts))
        .build();
```

- Pour plus de détails sur l'API, voir UploadPartla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios pour Amazon S3 utilisant des AWS SDK

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans Amazon S3 à l'aide de AWS kits SDK. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions dans Amazon S3. Chaque scénario inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter le code.

Exemples

- Création d'une URL présignée pour Amazon S3 à l'aide d'un SDK AWS
- Page Web répertoriant les objets Amazon S3 à l'aide d'un AWS SDK

- Supprimer les téléchargements partitionnés incomplets vers Amazon S3 à l'aide d'un SDK AWS
- Téléchargez tous les objets d'un compartiment Amazon Simple Storage Service (Amazon S3) dans un répertoire local.
- Obtenez un objet Amazon S3 à partir d'un point d'accès multirégional à l'aide d'un SDK AWS
- Obtenir un objet depuis un compartiment Amazon S3 à l'aide d'un AWS SDK, en spécifiant un en-tête If-Modified-Since
- Commencez à utiliser les buckets et les objets Amazon S3 à l'aide d'un SDK AWS
- Commencez à chiffrer les objets Amazon S3 à l'aide d'un AWS SDK
- Commencez à utiliser les balises pour les objets Amazon S3 à l'aide d'un AWS SDK
- Obtenez la configuration de conservation légale d'un objet Amazon S3 à l'aide d'un AWS SDK
- Utilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK
- Gérez les listes de contrôle d'accès (ACL) pour les compartiments Amazon S3 à l'aide d'un SDK AWS
- Gérez des objets Amazon S3 versionnés par lots à l'aide d'une fonction Lambda à l'aide d'un SDK AWS
- Analyser les URI Amazon S3 à l'aide d'un SDK AWS
- Réaliser une copie en plusieurs parties d'un objet Amazon S3 à l'aide d'un SDK AWS
- Effectuer un téléchargement partitionné d'un objet Amazon S3 à l'aide d'un SDK AWS
- Suivez le chargement ou le téléchargement d'un objet Amazon S3 à l'aide d'un AWS SDK
- Exemples d'approches pour les tests unitaires et d'intégration avec un AWS SDK
- Charger récursivement un répertoire local dans un compartiment Amazon Simple Storage Service (Amazon S3)
- Chargez ou téléchargez des fichiers volumineux vers et depuis Amazon S3 à l'aide d'un AWS SDK
- Chargez un flux de taille inconnue vers un objet Amazon S3 à l'aide d'un AWS SDK
- Utilisez des checksums pour travailler avec un objet Amazon S3 à l'aide d'un SDK AWS
- Travaillez avec des objets versionnés Amazon S3 à l'aide d'un SDK AWS

Création d'une URL présignée pour Amazon S3 à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment créer une URL présignée pour Amazon S3 et charger un objet.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Générez une URL présignée qui peut exécuter une action Amazon S3 pour une durée limitée.

```
    using System;
    using Amazon;
    using Amazon.S3;
    using Amazon.S3.Model;

    public class GenPresignedUrl
    {
        public static void Main()
        {
            const string bucketName = "doc-example-bucket";
            const string objectKey = "sample.txt";

            // Specify how long the presigned URL lasts, in hours
            const double timeoutDuration = 12;

            // Specify the AWS Region of your Amazon S3 bucket. If it is
            // different from the Region defined for the default user,
            // pass the Region to the constructor for the client. For
            // example: new AmazonS3Client(RegionEndpoint.USEast1);

            // If using the Region us-east-1, and server-side encryption with AWS
 KMS, you must specify Signature Version 4.
            // Region us-east-1 defaults to Signature Version 2 unless explicitly
 set to Version 4 as shown below.
            // For more details, see https://docs.aws.amazon.com/AmazonS3/latest/
userguide/UsingAWSSDK.html#specify-signature-version
            // and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/
TAWSConfigsS3.html
            AWSConfigsS3.UseSignatureVersion4 = true;
            IAmazonS3 s3Client = new AmazonS3Client(RegionEndpoint.USEast1);
```

```
            string urlString = GeneratePresignedURL(s3Client, bucketName,
objectKey, timeoutDuration);
            Console.WriteLine($"The generated URL is: {urlString}.");
        }

        /// <summary>
        /// Generate a presigned URL that can be used to access the file named
        /// in the objectKey parameter for the amount of time specified in the
        /// duration parameter.
        /// </summary>
        /// <param name="client">An initialized S3 client object used to call
        /// the GetPresignedUrl method.</param>
        /// <param name="bucketName">The name of the S3 bucket containing the
        /// object for which to create the presigned URL.</param>
        /// <param name="objectKey">The name of the object to access with the
        /// presigned URL.</param>
        /// <param name="duration">The length of time for which the presigned
        /// URL will be valid.</param>
        /// <returns>A string representing the generated presigned URL.</returns>
        public static string GeneratePresignedURL(IAmazonS3 client, string
bucketName, string objectKey, double duration)
        {
            string urlString = string.Empty;
            try
            {
                var request = new GetPreSignedUrlRequest()
                {
                    BucketName = bucketName,
                    Key = objectKey,
                    Expires = DateTime.UtcNow.AddHours(duration),
                };
                urlString = client.GetPreSignedURL(request);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error:'{ex.Message}'");
            }

            return urlString;
        }
    }
```

Générez une URL présignée et effectuez un chargement en utilisant cette URL.

```
    using System;
    using System.IO;
    using System.Net.Http;
    using System.Threading.Tasks;
    using Amazon;
    using Amazon.S3;
    using Amazon.S3.Model;

    /// <summary>
    /// This example shows how to upload an object to an Amazon Simple Storage
    /// Service (Amazon S3) bucket using a presigned URL. The code first
    /// creates a presigned URL and then uses it to upload an object to an
    /// Amazon S3 bucket using that URL.
    /// </summary>
    public class UploadUsingPresignedURL
    {
        private static HttpClient httpClient = new HttpClient();

        public static async Task Main()
        {
            string bucketName = "doc-example-bucket";
            string keyName = "samplefile.txt";
            string filePath = $"source\\{keyName}";

            // Specify how long the signed URL will be valid in hours.
            double timeoutDuration = 12;

            // Specify the AWS Region of your Amazon S3 bucket. If it is
            // different from the Region defined for the default user,
            // pass the Region to the constructor for the client. For
            // example: new AmazonS3Client(RegionEndpoint.USEast1);

            // If using the Region us-east-1, and server-side encryption with AWS
 KMS, you must specify Signature Version 4.
            // Region us-east-1 defaults to Signature Version 2 unless explicitly
 set to Version 4 as shown below.
            // For more details, see https://docs.aws.amazon.com/AmazonS3/latest/
userguide/UsingAWSSDK.html#specify-signature-version
            // and https://docs.aws.amazon.com/sdkfornet/v3/apidocs/items/Amazon/
TAWSConfigsS3.html
            AWSConfigsS3.UseSignatureVersion4 = true;
            IAmazonS3 client = new AmazonS3Client(RegionEndpoint.USEast1);
```

```
            var url = GeneratePreSignedURL(client, bucketName, keyName,
timeoutDuration);
            var success = await UploadObject(filePath, url);

            if (success)
            {
                Console.WriteLine("Upload succeeded.");
            }
            else
            {
                Console.WriteLine("Upload failed.");
            }
        }

        /// <summary>
        /// Uploads an object to an Amazon S3 bucket using the presigned URL
passed in
        /// the url parameter.
        /// </summary>
        /// <param name="filePath">The path (including file name) to the local
        /// file you want to upload.</param>
        /// <param name="url">The presigned URL that will be used to upload the
        /// file to the Amazon S3 bucket.</param>
        /// <returns>A Boolean value indicating the success or failure of the
        /// operation, based on the HttpWebResponse.</returns>
        public static async Task<bool> UploadObject(string filePath, string url)
        {
            using var streamContent = new StreamContent(
                new FileStream(filePath, FileMode.Open, FileAccess.Read));

            var response = await httpClient.PutAsync(url, streamContent);
            return response.IsSuccessStatusCode;
        }

        /// <summary>
        /// Generates a presigned URL which will be used to upload an object to
        /// an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// GetPreSignedURL.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket to which
the
```

```
        /// presigned URL will point.</param>
        /// <param name="objectKey">The name of the file that will be uploaded.</
param>
        /// <param name="duration">How long (in hours) the presigned URL will
        /// be valid.</param>
        /// <returns>The generated URL.</returns>
        public static string GeneratePreSignedURL(
            IAmazonS3 client,
            string bucketName,
            string objectKey,
            double duration)
        {
            var request = new GetPreSignedUrlRequest
            {
                BucketName = bucketName,
                Key = objectKey,
                Verb = HttpVerb.PUT,
                Expires = DateTime.UtcNow.AddHours(duration),
            };

            string url = client.GetPreSignedURL(request);
            return url;
        }
    }
```

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Générez une URL pré-signée pour télécharger un objet.

```
//! Routine which demonstrates creating a pre-signed URL to download an object
 from an
//! Amazon Simple Storage Service (Amazon S3) bucket.
```

```
/*!
  \param bucketName: Name of the bucket.
  \param key: Name of an object key.
  \param expirationSeconds: Expiration in seconds for pre-signed URL.
  \param clientConfig: Aws client configuration.
  \return Aws::String: A pre-signed URL.
*/
Aws::String AwsDoc::S3::generatePreSignedGetObjectUrl(const Aws::String
 &bucketName,
                                                       const Aws::String &key,
                                                       uint64_t expirationSeconds,
                                                       const
 Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    return client.GeneratePresignedUrl(bucketName, key,
 Aws::Http::HttpMethod::HTTP_GET,
                                       expirationSeconds);
}
```

Téléchargez à l'aide de libcurl.

```
static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
 *userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}

//! Utility routine to test getObject with a pre-signed URL.
/*!
  \param presignedURL: A pre-signed URL to get an object from a bucket.
  \param resultString: A string to hold the result.
  \return bool: Function succeeded.
*/
bool AwsDoc::S3::getObjectWithPresignedObjectUrl(const Aws::String &presignedURL,
                                                 Aws::String &resultString) {
    CURL *curl = curl_easy_init();
    CURLcode result;
```

```
    std::stringstream outWriteString;

    result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_URL" << std::endl;
        return false;
    }

    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }

    resultString = outWriteString.str();

    if (resultString.find("<?xml") == 0) {
        std::cerr << "Failed to get object, response:\n" << resultString <<
 std::endl;
        return false;
    }

    return true;
}
```

Générez une URL pré-signée pour télécharger un objet.

```
//! Routine which demonstrates creating a pre-signed URL to upload an object to
 an
//! Amazon Simple Storage Service (Amazon S3) bucket.
/*!
  \param bucketName: Name of the bucket.
  \param key: Name of an object key.
  \param clientConfig: Aws client configuration.
  \return Aws::String: A pre-signed URL.
*/
Aws::String AwsDoc::S3::generatePreSignedPutObjectUrl(const Aws::String
 &bucketName,
                                                      const Aws::String &key,
                                                      uint64_t expirationSeconds,
                                                      const
 Aws::S3::S3ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    return client.GeneratePresignedUrl(bucketName, key,
 Aws::Http::HttpMethod::HTTP_PUT,
                                       expirationSeconds);
}
```

Téléversez à l'aide de libcurl.

```
static size_t myCurlReadBack(char *buffer, size_t size, size_t nitems, void
 *userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    str->read(buffer, size * nitems);

    return str->gcount();
}

static size_t myCurlWriteBack(char *buffer, size_t size, size_t nitems, void
 *userdata) {
    Aws::StringStream *str = (Aws::StringStream *) userdata;

    if (nitems > 0) {
        str->write(buffer, size * nitems);
    }
    return size * nitems;
}
```

```
//! Utility routine to test putObject with a pre-signed URL.
/*!
  \param presignedURL: A pre-signed URL to put an object in a bucket.
  \param data: Body of the putObject request.
  \return bool: Function succeeded.
*/
bool AwsDoc::S3::PutStringWithPresignedObjectURL(const Aws::String &presignedURL,
                                                 const Aws::String &data) {
    CURL *curl = curl_easy_init();
    CURLcode result;

    Aws::StringStream readStringStream;
    readStringStream << data;
    result = curl_easy_setopt(curl, CURLOPT_READFUNCTION, myCurlReadBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_READFUNCTION" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_READDATA, &readStringStream);
    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_READDATA" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_INFILESIZE_LARGE,
                              (curl_off_t) data.size());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_INFILESIZE_LARGE" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, myCurlWriteBack);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEFUNCTION" << std::endl;
        return false;
    }

    std::stringstream outWriteString;

    result = curl_easy_setopt(curl, CURLOPT_WRITEDATA, &outWriteString);
```

```
    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_WRITEDATA " << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_URL, presignedURL.c_str());

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_URL" << std::endl;
        return false;
    }

    result = curl_easy_setopt(curl, CURLOPT_UPLOAD, 1L);

    if (result != CURLE_OK) {
        std::cerr << "Failed to set CURLOPT_PUT" << std::endl;
        return false;
    }

    result = curl_easy_perform(curl);

    if (result != CURLE_OK) {
        std::cerr << "Failed to perform CURL request" << std::endl;
        return false;
    }

    std::string outString = outWriteString.str();
    if (outString.empty()) {
        std::cout << "Successfully put object." << std::endl;
        return true;
    } else {
        std::cout << "A server error was encountered, output:\n" << outString
                  << std::endl;
        return false;
    }
}
```

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez des fonctions qui enveloppent les actions de présignature S3.

```
// Presigner encapsulates the Amazon Simple Storage Service (Amazon S3) presign
 actions
// used in the examples.
// It contains PresignClient, a client that is used to presign requests to Amazon
 S3.
// Presigned requests contain temporary credentials and can be made from any HTTP
 client.
type Presigner struct {
 PresignClient *s3.PresignClient
}



// GetObject makes a presigned request that can be used to get an object from a
 bucket.
// The presigned request is valid for the specified number of seconds.
func (presigner Presigner) GetObject(
 bucketName string, objectKey string, lifetimeSecs int64)
 (*v4.PresignedHTTPRequest, error) {
 request, err := presigner.PresignClient.PresignGetObject(context.TODO(),
 &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 }, func(opts *s3.PresignOptions) {
  opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
 })
 if err != nil {
  log.Printf("Couldn't get a presigned request to get %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
```

```
 return request, err
}



// PutObject makes a presigned request that can be used to put an object in a
 bucket.
// The presigned request is valid for the specified number of seconds.
func (presigner Presigner) PutObject(
 bucketName string, objectKey string, lifetimeSecs int64)
 (*v4.PresignedHTTPRequest, error) {
 request, err := presigner.PresignClient.PresignPutObject(context.TODO(),
 &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 }, func(opts *s3.PresignOptions) {
  opts.Expires = time.Duration(lifetimeSecs * int64(time.Second))
 })
 if err != nil {
  log.Printf("Couldn't get a presigned request to put %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return request, err
}



// DeleteObject makes a presigned request that can be used to delete an object
 from a bucket.
func (presigner Presigner) DeleteObject(bucketName string, objectKey string)
 (*v4.PresignedHTTPRequest, error) {
 request, err := presigner.PresignClient.PresignDeleteObject(context.TODO(),
 &s3.DeleteObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't get a presigned request to delete object %v. Here's why:
 %v\n", objectKey, err)
 }
 return request, err
}
```

Exécutez un exemple interactif qui génère et utilise des URL présignées pour charger, télécharger et supprimer un objet S3.

```
// RunPresigningScenario is an interactive example that shows you how to get
 presigned
// HTTP requests that you can use to move data into and out of Amazon Simple
 Storage
// Service (Amazon S3). The presigned requests contain temporary credentials and
 can
// be used by an HTTP client.
//
// 1. Get a presigned request to put an object in a bucket.
// 2. Use the net/http package to use the presigned request to upload a local
 file to the bucket.
// 3. Get a presigned request to get an object from a bucket.
// 4. Use the net/http package to use the presigned request to download the
 object to a local file.
// 5. Get a presigned request to delete an object from a bucket.
// 6. Use the net/http package to use the presigned request to delete the object.
//
// This example creates an Amazon S3 presign client from the specified sdkConfig
 so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
 example.
// This package can be found in the ..\..\demotools folder of this repo.
//
// It uses an IHttpRequester interface to abstract HTTP requests so they can be
 mocked
// during testing.
func RunPresigningScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner, httpRequester IHttpRequester) {
 defer func() {
  if r := recover(); r != nil {
   fmt.Printf("Something went wrong with the demo.")
  }
 }()

 log.Println(strings.Repeat("-", 88))
```

```
log.Println("Welcome to the Amazon S3 presigning demo.")
log.Println(strings.Repeat("-", 88))

s3Client := s3.NewFromConfig(sdkConfig)
bucketBasics := actions.BucketBasics{S3Client: s3Client}
presignClient := s3.NewPresignClient(s3Client)
presigner := actions.Presigner{PresignClient: presignClient}

bucketName := questioner.Ask("We'll need a bucket. Enter a name for a bucket "+
 "you own or one you want to create:", demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
if err != nil {
 panic(err)
}
if !bucketExists {
 err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
 if err != nil {
  panic(err)
 } else {
  log.Println("Bucket created.")
 }
}
log.Println(strings.Repeat("-", 88))

log.Printf("Let's presign a request to upload a file to your bucket.")
uploadFilename := questioner.Ask("Enter the path to a file you want to upload:",
 demotools.NotEmpty{})
uploadKey := questioner.Ask("What would you like to name the uploaded object?",
 demotools.NotEmpty{})
uploadFile, err := os.Open(uploadFilename)
if err != nil {
 panic(err)
}
defer uploadFile.Close()
presignedPutRequest, err := presigner.PutObject(bucketName, uploadKey, 60)
if err != nil {
 panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedPutRequest.Method,
 presignedPutRequest.URL)
log.Println("Using net/http to send the request...")
info, err := uploadFile.Stat()
if err != nil {
```

```
 panic(err)
}
putResponse, err := httpRequester.Put(presignedPutRequest.URL, info.Size(),
uploadFile)
if err != nil {
 panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
presignedPutRequest.Method,
 uploadKey, putResponse.StatusCode)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's presign a request to download the object.")
questioner.Ask("Press Enter when you're ready.")
presignedGetRequest, err := presigner.GetObject(bucketName, uploadKey, 60)
if err != nil {
 panic(err)
}
log.Printf("Got a presigned %v request to URL:\n\t%v\n",
presignedGetRequest.Method,
 presignedGetRequest.URL)
log.Println("Using net/http to send the request...")
getResponse, err := httpRequester.Get(presignedGetRequest.URL)
if err != nil {
 panic(err)
}
log.Printf("%v object %v with presigned URL returned %v.",
presignedGetRequest.Method,
 uploadKey, getResponse.StatusCode)
defer getResponse.Body.Close()
downloadBody, err := io.ReadAll(getResponse.Body)
if err != nil {
 panic(err)
}
log.Printf("Downloaded %v bytes. Here are the first 100 of them:\n",
len(downloadBody))
log.Println(strings.Repeat("-", 88))
log.Println(string(downloadBody[:100]))
log.Println(strings.Repeat("-", 88))

log.Println("Let's presign a request to delete the object.")
questioner.Ask("Press Enter when you're ready.")
presignedDelRequest, err := presigner.DeleteObject(bucketName, uploadKey)
if err != nil {
```

```
  panic(err)
 }
 log.Printf("Got a presigned %v request to URL:\n\t%v\n",
 presignedDelRequest.Method,
  presignedDelRequest.URL)
 log.Println("Using net/http to send the request...")
 delResponse, err := httpRequester.Delete(presignedDelRequest.URL)
 if err != nil {
  panic(err)
 }
 log.Printf("%v object %v with presigned URL returned %v.\n",
 presignedDelRequest.Method,
  uploadKey, delResponse.StatusCode)
 log.Println(strings.Repeat("-", 88))

 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Définissez un wrapper de requête HTTP utilisé par l'exemple pour effectuer des requêtes HTTP.

```
// IHttpRequester abstracts HTTP requests into an interface so it can be mocked
 during
// unit testing.
type IHttpRequester interface {
 Get(url string) (resp *http.Response, err error)
 Put(url string, contentLength int64, body io.Reader) (resp *http.Response, err
 error)
 Delete(url string) (resp *http.Response, err error)
}

// HttpRequester uses the net/http package to make HTTP requests during the
 scenario.
type HttpRequester struct{}

func (httpReq HttpRequester) Get(url string) (resp *http.Response, err error) {
 return http.Get(url)
}
```

```
func (httpReq HttpRequester) Put(url string, contentLength int64, body io.Reader)
 (resp *http.Response, err error) {
 putRequest, err := http.NewRequest("PUT", url, body)
 if err != nil {
  return nil, err
 }
 putRequest.ContentLength = contentLength
 return http.DefaultClient.Do(putRequest)
}
func (httpReq HttpRequester) Delete(url string) (resp *http.Response, err error)
 {
 delRequest, err := http.NewRequest("DELETE", url, nil)
 if err != nil {
  return nil, err
 }
 return http.DefaultClient.Do(delRequest)
}
```

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Générez une URL pré-signée pour un objet, puis téléchargez-le (requête GET).

Importations.

```
import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
 software.amazon.awssdk.services.s3.presigner.model.GetObjectPresignRequest;
import
 software.amazon.awssdk.services.s3.presigner.model.PresignedGetObjectRequest;
import software.amazon.awssdk.utils.IoUtils;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.file.Paths;
import java.time.Duration;
import java.util.UUID;
```

Générez l'URL.

```
   /* Create a pre-signed URL to download an object in a subsequent GET request.
*/
   public String createPresignedGetUrl(String bucketName, String keyName) {
       try (S3Presigner presigner = S3Presigner.create()) {

           GetObjectRequest objectRequest = GetObjectRequest.builder()
                   .bucket(bucketName)
                   .key(keyName)
                   .build();

           GetObjectPresignRequest presignRequest =
GetObjectPresignRequest.builder()
                   .signatureDuration(Duration.ofMinutes(10))  // The URL will
expire in 10 minutes.
                   .getObjectRequest(objectRequest)
                   .build();
```

```
            PresignedGetObjectRequest presignedRequest =
presigner.presignGetObject(presignRequest);
            logger.info("Presigned URL: [{}]",
presignedRequest.url().toString());
            logger.info("HTTP method: [{}]",
presignedRequest.httpRequest().method());

            return presignedRequest.url().toExternalForm();
        }
    }
```

Téléchargez l'objet en utilisant l'une des trois approches suivantes.

Utilisez la classe JDK `HttpURLConnection` (depuis v1.1) pour effectuer le téléchargement.

```
    /* Use the JDK HttpURLConnection (since v1.1) class to do the download. */
    public byte[] useHttpUrlConnectionToGet(String presignedUrlString) {
        ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.

        try {
            URL presignedUrl = new URL(presignedUrlString);
            HttpURLConnection connection = (HttpURLConnection)
presignedUrl.openConnection();
            connection.setRequestMethod("GET");
            // Download the result of executing the request.
            try (InputStream content = connection.getInputStream()) {
                IoUtils.copy(content, byteArrayOutputStream);
            }
            logger.info("HTTP response code is " + connection.getResponseCode());

        } catch (S3Exception | IOException e) {
            logger.error(e.getMessage(), e);
        }
        return byteArrayOutputStream.toByteArray();
    }
```

Utilisez la classe JDK `HttpClient` (depuis la version 11) pour effectuer le téléchargement.

```
    /* Use the JDK HttpClient (since v11) class to do the download. */
```

```
   public byte[] useHttpClientToGet(String presignedUrlString) {
       ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.

       HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
       HttpClient httpClient = HttpClient.newHttpClient();
       try {
           URL presignedUrl = new URL(presignedUrlString);
           HttpResponse<InputStream> response = httpClient.send(requestBuilder
                           .uri(presignedUrl.toURI())
                           .GET()
                           .build(),
                   HttpResponse.BodyHandlers.ofInputStream());

           IoUtils.copy(response.body(), byteArrayOutputStream);

           logger.info("HTTP response code is " + response.statusCode());

       } catch (URISyntaxException | InterruptedException | IOException e) {
           logger.error(e.getMessage(), e);
       }
       return byteArrayOutputStream.toByteArray();
   }
```

Utilisez la classe AWS SDK for `SdkHttpClient` Java pour effectuer le téléchargement.

```
   /* Use the AWS SDK for Java SdkHttpClient class to do the download. */
   public byte[] useSdkHttpClientToPut(String presignedUrlString) {

       ByteArrayOutputStream byteArrayOutputStream = new
ByteArrayOutputStream(); // Capture the response body to a byte array.
       try {
           URL presignedUrl = new URL(presignedUrlString);
           SdkHttpRequest request = SdkHttpRequest.builder()
                   .method(SdkHttpMethod.GET)
                   .uri(presignedUrl.toURI())
                   .build();

           HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
                   .request(request)
                   .build();
```

```
            try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
                HttpExecuteResponse response =
 sdkHttpClient.prepareRequest(executeRequest).call();
                response.responseBody().ifPresentOrElse(
                        abortableInputStream -> {
                            try {
                                IoUtils.copy(abortableInputStream,
 byteArrayOutputStream);
                            } catch (IOException e) {
                                throw new RuntimeException(e);
                            }
                        },
                        () -> logger.error("No response body."));

                logger.info("HTTP Response code is {}",
 response.httpResponse().statusCode());
            }
        } catch (URISyntaxException | IOException e) {
            logger.error(e.getMessage(), e);
        }
        return byteArrayOutputStream.toByteArray();
    }
```

Générez une URL pré-signée pour un téléchargement, puis chargez un fichier (requête PUT).

Importations.

```
import com.example.s3.util.PresignUrlUtils;
import org.slf4j.Logger;
import software.amazon.awssdk.core.internal.sync.FileContentStreamProvider;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.presigner.S3Presigner;
import
 software.amazon.awssdk.services.s3.presigner.model.PresignedPutObjectRequest;
```

```
import
 software.amazon.awssdk.services.s3.presigner.model.PutObjectPresignRequest;

import java.io.File;
import java.io.IOException;
import java.io.OutputStream;
import java.io.RandomAccessFile;
import java.net.HttpURLConnection;
import java.net.URISyntaxException;
import java.net.URL;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.ByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Duration;
import java.util.Map;
import java.util.UUID;
```

Générez l'URL.

```
   /* Create a presigned URL to use in a subsequent PUT request */
   public String createPresignedUrl(String bucketName, String keyName,
Map<String, String> metadata) {
       try (S3Presigner presigner = S3Presigner.create()) {

           PutObjectRequest objectRequest = PutObjectRequest.builder()
                   .bucket(bucketName)
                   .key(keyName)
                   .metadata(metadata)
                   .build();

           PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
                   .signatureDuration(Duration.ofMinutes(10))  // The URL
expires in 10 minutes.
                   .putObjectRequest(objectRequest)
                   .build();
```

```
            PresignedPutObjectRequest presignedRequest =
 presigner.presignPutObject(presignRequest);
            String myURL = presignedRequest.url().toString();
            logger.info("Presigned URL to upload a file to: [{}]", myURL);
            logger.info("HTTP method: [{}]",
 presignedRequest.httpRequest().method());

            return presignedRequest.url().toExternalForm();
        }
    }
```

Téléchargez un objet de fichier en utilisant l'une des trois approches suivantes.

Utilisez la classe JDK `HttpURLConnection` (depuis la version 1.1) pour effectuer le téléchargement.

```
    /* Use the JDK HttpURLConnection (since v1.1) class to do the upload. */
    public void useHttpUrlConnectionToPut(String presignedUrlString, File
 fileToPut, Map<String, String> metadata) {
        logger.info("Begin [{}] upload", fileToPut.toString());
        try {
            URL presignedUrl = new URL(presignedUrlString);
            HttpURLConnection connection = (HttpURLConnection)
 presignedUrl.openConnection();
            connection.setDoOutput(true);
            metadata.forEach((k, v) -> connection.setRequestProperty("x-amz-
meta-" + k, v));
            connection.setRequestMethod("PUT");
            OutputStream out = connection.getOutputStream();

            try (RandomAccessFile file = new RandomAccessFile(fileToPut, "r");
                 FileChannel inChannel = file.getChannel()) {
                ByteBuffer buffer = ByteBuffer.allocate(8192); //Buffer size is
 8k

                while (inChannel.read(buffer) > 0) {
                    buffer.flip();
                    for (int i = 0; i < buffer.limit(); i++) {
                        out.write(buffer.get());
                    }
                    buffer.clear();
                }
```

```
            } catch (IOException e) {
                logger.error(e.getMessage(), e);
            }

            out.close();
            connection.getResponseCode();
            logger.info("HTTP response code is " + connection.getResponseCode());

        } catch (S3Exception | IOException e) {
            logger.error(e.getMessage(), e);
        }
    }
```

Utilisez la classe JDK `HttpClient` (depuis la version 11) pour effectuer le téléchargement.

```
    /* Use the JDK HttpClient (since v11) class to do the upload. */
    public void useHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
        logger.info("Begin [{}] upload", fileToPut.toString());

        HttpRequest.Builder requestBuilder = HttpRequest.newBuilder();
        metadata.forEach((k, v) -> requestBuilder.header("x-amz-meta-" + k, v));

        HttpClient httpClient = HttpClient.newHttpClient();
        try {
            final HttpResponse<Void> response = httpClient.send(requestBuilder
                            .uri(new URL(presignedUrlString).toURI())

.PUT(HttpRequest.BodyPublishers.ofFile(Path.of(fileToPut.toURI())))
                            .build(),
                    HttpResponse.BodyHandlers.discarding());

            logger.info("HTTP response code is " + response.statusCode());

        } catch (URISyntaxException | InterruptedException | IOException e) {
            logger.error(e.getMessage(), e);
        }
    }
```

Utilisez la `SdkHttpClient` classe AWS for Java V2 pour effectuer le téléchargement.

```
   /* Use the AWS SDK for Java V2 SdkHttpClient class to do the upload. */
   public void useSdkHttpClientToPut(String presignedUrlString, File fileToPut,
Map<String, String> metadata) {
       logger.info("Begin [{}] upload", fileToPut.toString());

       try {
           URL presignedUrl = new URL(presignedUrlString);

           SdkHttpRequest.Builder requestBuilder = SdkHttpRequest.builder()
                   .method(SdkHttpMethod.PUT)
                   .uri(presignedUrl.toURI());
           // Add headers
           metadata.forEach((k, v) -> requestBuilder.putHeader("x-amz-meta-" +
k, v));
           // Finish building the request.
           SdkHttpRequest request = requestBuilder.build();

           HttpExecuteRequest executeRequest = HttpExecuteRequest.builder()
                   .request(request)
                   .contentStreamProvider(new
FileContentStreamProvider(fileToPut.toPath()))
                   .build();

           try (SdkHttpClient sdkHttpClient = ApacheHttpClient.create()) {
               HttpExecuteResponse response =
sdkHttpClient.prepareRequest(executeRequest).call();
               logger.info("Response code: {}",
response.httpResponse().statusCode());
           }
       } catch (URISyntaxException | IOException e) {
           logger.error(e.getMessage(), e);
       }
   }
```

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une URL présignée pour charger un objet dans un compartiment.

```
import https from "https";
import { PutObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
  getSignedUrl,
  S3RequestPresigner,
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";

const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(
    new HttpRequest({ ...url, method: "PUT" }),
  );
  return formatUrl(signedUrlObject);
};

const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
  const command = new PutObjectCommand({ Bucket: bucket, Key: key });
  return getSignedUrl(client, command, { expiresIn: 3600 });
};
```

```
function put(url, data) {
  return new Promise((resolve, reject) => {
    const req = https.request(
      url,
      { method: "PUT", headers: { "Content-Length": new Blob([data]).size } },
      (res) => {
        let responseBody = "";
        res.on("data", (chunk) => {
          responseBody += chunk;
        });
        res.on("end", () => {
          resolve(responseBody);
        });
      },
    );
    req.on("error", (err) => {
      reject(err);
    });
    req.write(data);
    req.end();
  });
}

export const main = async () => {
  const REGION = "us-east-1";
  const BUCKET = "example_bucket";
  const KEY = "example_file.txt";

  // There are two ways to generate a presigned URL.
  // 1. Use createPresignedUrl without the S3 client.
  // 2. Use getSignedUrl in conjunction with the S3 client and GetObjectCommand.
  try {
    const noClientUrl = await createPresignedUrlWithoutClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    const clientUrl = await createPresignedUrlWithClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });
```

```
    // After you get the presigned URL, you can provide your own file
    // data. Refer to put() above.
    console.log("Calling PUT using presigned URL without client");
    await put(noClientUrl, "Hello World");

    console.log("Calling PUT using presigned URL with client");
    await put(clientUrl, "Hello World");

    console.log("\nDone. Check your S3 console.");
  } catch (err) {
    console.error(err);
  }
};
```

Créez une URL présignée pour télécharger un objet à partir d'un compartiment.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { fromIni } from "@aws-sdk/credential-providers";
import { HttpRequest } from "@smithy/protocol-http";
import {
  getSignedUrl,
  S3RequestPresigner,
} from "@aws-sdk/s3-request-presigner";
import { parseUrl } from "@smithy/url-parser";
import { formatUrl } from "@aws-sdk/util-format-url";
import { Hash } from "@smithy/hash-node";

const createPresignedUrlWithoutClient = async ({ region, bucket, key }) => {
  const url = parseUrl(`https://${bucket}.s3.${region}.amazonaws.com/${key}`);
  const presigner = new S3RequestPresigner({
    credentials: fromIni(),
    region,
    sha256: Hash.bind(null, "sha256"),
  });

  const signedUrlObject = await presigner.presign(new HttpRequest(url));
  return formatUrl(signedUrlObject);
};

const createPresignedUrlWithClient = ({ region, bucket, key }) => {
  const client = new S3Client({ region });
```

```
  const command = new GetObjectCommand({ Bucket: bucket, Key: key });
  return getSignedUrl(client, command, { expiresIn: 3600 });
};

export const main = async () => {
  const REGION = "us-east-1";
  const BUCKET = "example_bucket";
  const KEY = "example_file.jpg";

  try {
    const noClientUrl = await createPresignedUrlWithoutClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    const clientUrl = await createPresignedUrlWithClient({
      region: REGION,
      bucket: BUCKET,
      key: KEY,
    });

    console.log("Presigned URL without client");
    console.log(noClientUrl);
    console.log("\n");

    console.log("Presigned URL with client");
    console.log(clientUrl);
  } catch (err) {
    console.error(err);
  }
};
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une demande présignée `GetObject` et utilisez l'URL pour télécharger un objet.

```
suspend fun getObjectPresigned(
    s3: S3Client,
    bucketName: String,
    keyName: String,
): String {
    // Create a GetObjectRequest.
    val unsignedRequest =
        GetObjectRequest {
            bucket = bucketName
            key = keyName
        }

    // Presign the GetObject request.
    val presignedRequest = s3.presignGetObject(unsignedRequest, 24.hours)

    // Use the URL from the presigned HttpRequest in a subsequent HTTP GET
 request to retrieve the object.
    val objectContents = URL(presignedRequest.url.toString()).readText()

    return objectContents
}
```

Créez une demande `GetObject` présignée avec des options avancées.

```
suspend fun getObjectPresignedMoreOptions(
    s3: S3Client,
    bucketName: String,
    keyName: String,
): HttpRequest {
```

```
    // Create a GetObjectRequest.
    val unsignedRequest =
        GetObjectRequest {
            bucket = bucketName
            key = keyName
        }

    // Presign the GetObject request.
    val presignedRequest =
        s3.presignGetObject(unsignedRequest, signer = CrtAwsSigner) {
            signingDate = Instant.now() + 12.hours // Presigned request can be
 used 12 hours from now.
            algorithm = AwsSigningAlgorithm.SIGV4_ASYMMETRIC
            signatureType = AwsSignatureType.HTTP_REQUEST_VIA_QUERY_PARAMS
            expiresAfter = 8.hours // Presigned request expires 8 hours later.
        }
    return presignedRequest
}
```

Créez une demande présignée `PutObject` et utilisez-la pour charger un objet.

```
suspend fun putObjectPresigned(
    s3: S3Client,
    bucketName: String,
    keyName: String,
    content: String,
) {
    // Create a PutObjectRequest.
    val unsignedRequest =
        PutObjectRequest {
            bucket = bucketName
            key = keyName
        }

    // Presign the request.
    val presignedRequest = s3.presignPutObject(unsignedRequest, 24.hours)

    // Use the URL and any headers from the presigned HttpRequest in a subsequent
 HTTP PUT request to retrieve the object.
    // Create a PUT request using the OKHttpClient API.
    val putRequest =
        Request
```

```
            .Builder()
            .url(presignedRequest.url.toString())
            .apply {
                presignedRequest.headers.forEach { key, values ->
                    header(key, values.joinToString(", "))
                }
            }.put(content.toRequestBody())
            .build()

    val response = OkHttpClient().newCall(putRequest).execute()
    assert(response.isSuccessful)
}
```

- Pour en savoir plus, consultez Guide du développeur d'AWS SDK pour Kotlin.

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
namespace S3;
use Aws\Exception\AwsException;
use AwsUtilities\PrintableLineBreak;
use AwsUtilities\TestableReadline;
use DateTime;

require 'vendor/autoload.php';

class PresignedURL
{
    use PrintableLineBreak;
    use TestableReadline;

    public function run()
    {
        $s3Service = new S3Service();
```

```
        $expiration = new DateTime("+20 minutes");
        $linebreak = $this->getLineBreak();

        echo $linebreak;
        echo ("Welcome to the Amazon S3 presigned URL demo.\n");
        echo $linebreak;

        $bucket = $this->testable_readline("First, please enter the name of the
 S3 bucket to use: ");
        $key = $this->testable_readline("Next, provide the key of an object in
 the given bucket: ");
        echo $linebreak;
        $command = $s3Service->getClient()->getCommand('GetObject', [
            'Bucket' => $bucket,
            'Key' => $key,
        ]);
        try {
            $preSignedUrl = $s3Service->preSignedUrl($command, $expiration);
            echo "Your preSignedUrl is \n$preSignedUrl\nand will be good for the
 next 20 minutes.\n";
            echo $linebreak;
            echo "Thanks for trying the Amazon S3 presigned URL demo.\n";
        } catch (AwsException $exception) {
            echo $linebreak;
            echo "Something went wrong: $exception";
            die();
        }
    }
}

$runner = new PresignedURL();
$runner->run();
```

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Générez une URL présignée qui peut exécuter une action S3 pour une durée limitée. Utilisez le package Requests (Requêtes) pour effectuer une requête avec l'URL.

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)


def generate_presigned_url(s3_client, client_method, method_parameters,
 expires_in):
    """
    Generate a presigned Amazon S3 URL that can be used to perform an action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method, Params=method_parameters,
 ExpiresIn=expires_in
        )
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.", client_method
```

```
        )
        raise
    return url


def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Welcome to the Amazon S3 presigned URL demo.")
    print("-" * 88)

    parser = argparse.ArgumentParser()
    parser.add_argument("bucket", help="The name of the bucket.")
    parser.add_argument(
        "key",
        help="For a GET operation, the key of the object in Amazon S3. For a "
        "PUT operation, the name of a file to upload.",
    )
    parser.add_argument("action", choices=("get", "put"), help="The action to
 perform.")
    args = parser.parse_args()

    s3_client = boto3.client("s3")
    client_action = "get_object" if args.action == "get" else "put_object"
    url = generate_presigned_url(
        s3_client, client_action, {"Bucket": args.bucket, "Key": args.key}, 1000
    )

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == "get":
        response = requests.get(url)
    elif args.action == "put":
        print("Putting data to the URL.")
        try:
            with open(args.key, "r") as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
            print(
                f"Couldn't find {args.key}. For a PUT operation, the key must be
 the "
                f"name of a file that exists on your computer."
```

```
            )

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print("-" * 88)


if __name__ == "__main__":
    usage_demo()
```

Générez une requête POST présignée pour charger un fichier.

```
class BucketWrapper:
    """Encapsulates S3 bucket actions."""

    def __init__(self, bucket):
        """
        :param bucket: A Boto3 Bucket resource. This is a high-level resource in
 Boto3
                       that wraps bucket actions in a class-like structure.
        """
        self.bucket = bucket
        self.name = bucket.name


    def generate_presigned_post(self, object_key, expires_in):
        """
        Generate a presigned Amazon S3 POST request to upload a file.
        A presigned POST can be used for a limited time to let someone without an
 AWS
        account upload a file to a bucket.

        :param object_key: The object key to identify the uploaded object.
        :param expires_in: The number of seconds the presigned POST is valid.
        :return: A dictionary that contains the URL and form fields that contain
                 required access data.
        """
        try:
            response = self.bucket.meta.client.generate_presigned_post(
```

```
                Bucket=self.bucket.name, Key=object_key, ExpiresIn=expires_in
            )
            logger.info("Got presigned POST URL: %s", response["url"])
        except ClientError:
            logger.exception(
                "Couldn't get a presigned POST URL for bucket '%s' and object
 '%s'",
                self.bucket.name,
                object_key,
            )
            raise
        return response
```

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"
require "net/http"

# Creates a presigned URL that can be used to upload content to an object.
#
# @param bucket [Aws::S3::Bucket] An existing Amazon S3 bucket.
# @param object_key [String] The key to give the uploaded object.
# @return [URI, nil] The parsed URI if successful; otherwise nil.
def get_presigned_url(bucket, object_key)
  url = bucket.object(object_key).presigned_url(:put)
  puts "Created presigned URL: #{url}"
  URI(url)
rescue Aws::Errors::ServiceError => e
  puts "Couldn't create presigned URL for #{bucket.name}:#{object_key}. Here's
 why: #{e.message}"
```

```
end

# Example usage:
def run_demo
  bucket_name = "doc-example-bucket"
  object_key = "my-file.txt"
  object_content = "This is the content of my-file.txt."

  bucket = Aws::S3::Bucket.new(bucket_name)
  presigned_url = get_presigned_url(bucket, object_key)
  return unless presigned_url

  response = Net::HTTP.start(presigned_url.host) do |http|
    http.send_request("PUT", presigned_url.request_uri, object_content,
 "content_type" => "")
  end

  case response
  when Net::HTTPSuccess
    puts "Content uploaded!"
  else
    puts response.value
  end
end

run_demo if $PROGRAM_NAME == __FILE__
```

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez des demandes de présignature pour les objets S3 GET et PUT.

```
async fn get_object(
    client: &Client,
```

```
    bucket: &str,
    object: &str,
    expires_in: u64,
) -> Result<(), Box<dyn Error>> {
    let expires_in = Duration::from_secs(expires_in);
    let presigned_request = client
        .get_object()
        .bucket(bucket)
        .key(object)
        .presigned(PresigningConfig::expires_in(expires_in)?)
        .await?;

    println!("Object URI: {}", presigned_request.uri());

    Ok(())
}

async fn put_object(
    client: &Client,
    bucket: &str,
    object: &str,
    expires_in: u64,
) -> Result<(), Box<dyn Error>> {
    let expires_in = Duration::from_secs(expires_in);

    let presigned_request = client
        .put_object()
        .bucket(bucket)
        .key(object)
        .presigned(PresigningConfig::expires_in(expires_in)?)
        .await?;

    println!("Object URI: {}", presigned_request.uri());

    Ok(())
}
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Page Web répertoriant les objets Amazon S3 à l'aide d'un AWS SDK

L'exemple de code suivant montre comment répertorier des objets Amazon S3 dans une page web.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Le code suivant est le composant React pertinent qui appelle le AWS SDK. Une version exécutable de l'application contenant ce composant se trouve sur le lien précédent GitHub .

```
import { useEffect, useState } from "react";
import {
  ListObjectsCommand,
  ListObjectsCommandOutput,
  S3Client,
} from "@aws-sdk/client-s3";
import { fromCognitoIdentityPool } from "@aws-sdk/credential-providers";
import "./App.css";

function App() {
  const [objects, setObjects] = useState<
    Required<ListObjectsCommandOutput>["Contents"]
  >([]);

  useEffect(() => {
    const client = new S3Client({
      region: "us-east-1",
      // Unless you have a public bucket, you'll need access to a private bucket.
      // One way to do this is to create an Amazon Cognito identity pool, attach
 a role to the pool,
      // and grant the role access to the 's3:GetObject' action.
      //
      // You'll also need to configure the CORS settings on the bucket to allow
 traffic from
```

```
      // this example site. Here's an example configuration that allows all
 origins. Don't
      // do this in production.
      //[
      //  {
      //    "AllowedHeaders": ["*"],
      //    "AllowedMethods": ["GET"],
      //    "AllowedOrigins": ["*"],
      //    "ExposeHeaders": [],
      //  },
      //]
      //
      credentials: fromCognitoIdentityPool({
        clientConfig: { region: "us-east-1" },
        identityPoolId: "<YOUR_IDENTITY_POOL_ID>",
      }),
    });
    const command = new ListObjectsCommand({ Bucket: "bucket-name" });
    client.send(command).then(({ Contents }) => setObjects(Contents || []));
  }, []);

  return (
    <div className="App">
      {objects.map((o) => (
        <div key={o.ETag}>{o.Key}</div>
      ))}
    </div>
  );
}

export default App;
```

- Pour plus de détails sur l'API, reportez-vous ListObjectsà la section Référence des AWS SDK for JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Supprimer les téléchargements partitionnés incomplets vers Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment supprimer ou arrêter les téléchargements partitionnés incomplets sur Amazon S3.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Pour arrêter les téléchargements partitionnés en cours ou incomplets pour une quelconque raison, vous pouvez obtenir une liste des téléchargements, puis les supprimer comme indiqué dans l'exemple suivant.

```
   public static void abortIncompleteMultipartUploadsFromList() {
       ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
               .bucket(bucketName)
               .build();

       ListMultipartUploadsResponse response =
s3Client.listMultipartUploads(listMultipartUploadsRequest);
       List<MultipartUpload> uploads = response.uploads();

       AbortMultipartUploadRequest abortMultipartUploadRequest;
       for (MultipartUpload upload : uploads) {
           abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                   .bucket(bucketName)
                   .key(upload.key())
                   .expectedBucketOwner(accountId)
                   .uploadId(upload.uploadId())
                   .build();

           AbortMultipartUploadResponse abortMultipartUploadResponse =
s3Client.abortMultipartUpload(abortMultipartUploadRequest);
```

```
            if (abortMultipartUploadResponse.sdkHttpResponse().isSuccessful()) {
                logger.info("Upload ID [{}] to bucket [{}] successfully
aborted.", upload.uploadId(), bucketName);
            }
        }
    }
```

Pour supprimer des téléchargements partitionnés incomplets initiés avant ou après une date, vous pouvez supprimer les téléchargements partitionnés de manière sélective en fonction d'un moment donné, comme indiqué dans l'exemple suivant.

```
    static void abortIncompleteMultipartUploadsOlderThan(Instant pointInTime) {
        ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
                .bucket(bucketName)
                .build();

        ListMultipartUploadsResponse response =
s3Client.listMultipartUploads(listMultipartUploadsRequest);
        List<MultipartUpload> uploads = response.uploads();

        AbortMultipartUploadRequest abortMultipartUploadRequest;
        for (MultipartUpload upload : uploads) {
            logger.info("Found multipartUpload with upload ID [{}], initiated
[{}]", upload.uploadId(), upload.initiated());
            if (upload.initiated().isBefore(pointInTime)) {
                abortMultipartUploadRequest =
AbortMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(upload.key())
                        .expectedBucketOwner(accountId)
                        .uploadId(upload.uploadId())
                        .build();

                AbortMultipartUploadResponse abortMultipartUploadResponse =
s3Client.abortMultipartUpload(abortMultipartUploadRequest);
                if
(abortMultipartUploadResponse.sdkHttpResponse().isSuccessful()) {
                    logger.info("Upload ID [{}] to bucket [{}] successfully
aborted.", upload.uploadId(), bucketName);
                }
            }
```

```
        }
    }
```

Si vous avez accès à l'identifiant de téléchargement après avoir commencé un téléchargement en plusieurs parties, vous pouvez supprimer le téléchargement en cours à l'aide de cet identifiant.

```
    static void abortMultipartUploadUsingUploadId() {
        String uploadId = startUploadReturningUploadId();
        AbortMultipartUploadResponse response = s3Client.abortMultipartUpload(b -
> b
                .uploadId(uploadId)
                .bucket(bucketName)
                .key(key));

        if (response.sdkHttpResponse().isSuccessful()) {
            logger.info("Upload ID [{}] to bucket [{}] successfully aborted.",
 uploadId, bucketName);
        }
    }
```

Pour supprimer systématiquement les téléchargements partitionnés incomplets datant de plus d'un certain nombre de jours, configurez une configuration du cycle de vie du bucket pour le bucket. L'exemple suivant montre comment créer une règle pour supprimer les téléchargements incomplets datant de plus de 7 jours.

```
    static void abortMultipartUploadsUsingLifecycleConfig() {
        Collection<LifecycleRule> lifeCycleRules =
 List.of(LifecycleRule.builder()
                .abortIncompleteMultipartUpload(b -> b.
                        daysAfterInitiation(7))
                .status("Enabled")
                .filter(SdkBuilder::build) // Filter element is required.
                .build());

        // If the action is successful, the service sends back an HTTP 200
 response with an empty HTTP body.
        PutBucketLifecycleConfigurationResponse response =
 s3Client.putBucketLifecycleConfiguration(b -> b
                .bucket(bucketName)
```

```
                .lifecycleConfiguration(b1 -> b1.rules(lifeCycleRules)));

        if (response.sdkHttpResponse().isSuccessful()) {
            logger.info("Rule to abort incomplete multipart uploads added to
bucket.");
        } else {
            logger.error("Unsuccessfully applied rule. HTTP status code is [{}]",
response.sdkHttpResponse().statusCode());
        }
    }
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - AbortMultipartUpload
  - ListMultipartUploads
  - PutBucketLifecycleConfiguration

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Téléchargez tous les objets d'un compartiment Amazon Simple Storage Service (Amazon S3) dans un répertoire local.

L'exemple de code suivant montre comment télécharger tous les objets d'un compartiment Amazon Simple Storage Service (Amazon S3) dans un répertoire local.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Utilisez un S3 TransferManager pour télécharger tous les objets S3 dans le même compartiment S3. Consultez le fichier complet et le test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DirectoryDownload;
import software.amazon.awssdk.transfer.s3.model.DownloadDirectoryRequest;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.HashSet;
import java.util.Set;
import java.util.UUID;
import java.util.stream.Collectors;

    public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
            URI destinationPathURI, String bucketName) {
        DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
                .destination(Paths.get(destinationPathURI))
                .bucket(bucketName)
                .build());
        CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

        completedDirectoryDownload.failedTransfers()
                .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
        return completedDirectoryDownload.failedTransfers().size();
    }
```

- Pour plus de détails sur l'API, reportez-vous DownloadDirectoryà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Obtenez un objet Amazon S3 à partir d'un point d'accès multirégional à l'aide d'un SDK AWS

L'exemple de code suivant montre comment obtenir un objet à partir d'un point d'accès multirégional.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Configurez le client S3 pour utiliser l'algorithme de signature asymétrique Sigv4 (Sigv4a).

```
        suspend fun createS3Client(): S3Client {
            // Configure your S3Client to use the Asymmetric Sigv4 (Sigv4a)
signing algorithm.
            val sigV4AScheme = SigV4AsymmetricAuthScheme(CrtAwsSigner)
            val s3 = S3Client.fromEnvironment {
                authSchemes = listOf(sigV4AScheme)
            }
            return s3
        }
```

Utilisez l'ARN du point d'accès multirégional au lieu d'un nom de compartiment pour récupérer l'objet.

```
    suspend fun getObjectFromMrap(
        s3: S3Client,
        mrapArn: String,
        keyName: String,
    ): String? {
```

```
        val request = GetObjectRequest {
            bucket = mrapArn // Use the ARN instead of the bucket name for object
operations.
            key = keyName
        }

        var stringObj: String? = null
        s3.getObject(request) { resp ->
            stringObj = resp.body?.decodeToString()
            if (stringObj != null) {
                println("Successfully read $keyName from $mrapArn")
            }
        }
        return stringObj
    }
```

- Pour en savoir plus, consultez Guide du développeur d'AWS SDK pour Kotlin.
- Pour plus de détails sur l'API, reportez-vous GetObjectà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Obtenir un objet depuis un compartiment Amazon S3 à l'aide d'un AWS SDK, en spécifiant un en-tête If-Modified-Since

L'exemple de code suivant montre comment lire les données d'un objet dans un compartiment S3, mais uniquement si ce compartiment n'a pas été modifié depuis la dernière extraction.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
use aws_sdk_s3::{
    error::SdkError,
    operation::head_object::HeadObjectError,
    primitives::{ByteStream, DateTime, DateTimeFormat},
    Client, Error,
};
use tracing::{error, warn};

const KEY: &str = "key";
const BODY: &str = "Hello, world!";

/// Demonstrate how `if-modified-since` reports that matching objects haven't
/// changed.
///
/// # Steps
/// - Create a bucket.
/// - Put an object in the bucket.
/// - Get the bucket headers.
/// - Get the bucket headers again but only if modified.
/// - Delete the bucket.
#[tokio::main]
async fn main() -> Result<(), Error> {
    tracing_subscriber::fmt::init();

    // Get a new UUID to use when creating a unique bucket name.
    let uuid = uuid::Uuid::new_v4();

    // Load the AWS configuration from the environment.
    let client = Client::new(&aws_config::load_from_env().await);

    // Generate a unique bucket name using the previously generated UUID.
    // Then create a new bucket with that name.
    let bucket_name = format!("if-modified-since-{uuid}");
    client
        .create_bucket()
        .bucket(bucket_name.clone())
        .send()
        .await?;

    // Create a new object in the bucket whose name is `KEY` and whose
    // contents are `BODY`.
    let put_object_output = client
        .put_object()
```

```
        .bucket(bucket_name.as_str())
        .key(KEY)
        .body(ByteStream::from_static(BODY.as_bytes()))
        .send()
        .await;

    // If the `PutObject` succeeded, get the eTag string from it. Otherwise,
    // report an error and return an empty string.
    let e_tag_1 = match put_object_output {
        Ok(put_object) => put_object.e_tag.unwrap(),
        Err(err) => {
            error!("{err:?}");
            String::new()
        }
    };

    // Request the object's headers.
    let head_object_output = client
        .head_object()
        .bucket(bucket_name.as_str())
        .key(KEY)
        .send()
        .await;

    // If the `HeadObject` request succeeded, create a tuple containing the
    // values of the headers `last-modified` and `etag`. If the request
    // failed, return the error in a tuple instead.
    let (last_modified, e_tag_2) = match head_object_output {
        Ok(head_object) => (
            Ok(head_object.last_modified().cloned().unwrap()),
            head_object.e_tag.unwrap(),
        ),
        Err(err) => (Err(err), String::new()),
    };

    warn!("last modified: {last_modified:?}");
    assert_eq!(
        e_tag_1, e_tag_2,
        "PutObject and first GetObject had differing eTags"
    );

    println!("First value of last_modified: {last_modified:?}");
    println!("First tag: {}\n", e_tag_1);
```

```
    // Send a second `HeadObject` request. This time, the `if_modified_since`
    // option is specified, giving the `last_modified` value returned by the
    // first call to `HeadObject`.
    //
    // Since the object hasn't been changed, and there are no other objects in
    // the bucket, there should be no matching objects.

    let head_object_output = client
        .head_object()
        .bucket(bucket_name.as_str())
        .key(KEY)
        .if_modified_since(last_modified.unwrap())
        .send()
        .await;

    // If the `HeadObject` request succeeded, the result is a typle containing
    // the `last_modified` and `e_tag_1` properties. This is _not_ the expected
    // result.
    //
    // The _expected_ result of the second call to `HeadObject` is an
    // `SdkError::ServiceError` containing the HTTP error response. If that's
    // the case and the HTTP status is 304 (not modified), the output is a
    // tuple containing the values of the HTTP `last-modified` and `etag`
    // headers.
    //
    // If any other HTTP error occurred, the error is returned as an
    // `SdkError::ServiceError`.

    let (last_modified, e_tag_2): (Result<DateTime, SdkError<HeadObjectError>>,
String) =
        match head_object_output {
            Ok(head_object) => (
                Ok(head_object.last_modified().cloned().unwrap()),
                head_object.e_tag.unwrap(),
            ),
            Err(err) => match err {
                SdkError::ServiceError(err) => {
                    // Get the raw HTTP response. If its status is 304, the
                    // object has not changed. This is the expected code path.
                    let http = err.raw();
                    match http.status().as_u16() {
                        // If the HTTP status is 304: Not Modified, return a
                        // tuple containing the values of the HTTP
                        // `last-modified` and `etag` headers.
```

```
                        304 => (
                            Ok(DateTime::from_str(
                                http.headers().get("last-modified").unwrap(),
                                DateTimeFormat::HttpDate,
                            )
                            .unwrap()),
                            http.headers().get("etag").map(|t|
t.into()).unwrap(),
                        ),
                        // Any other HTTP status code is returned as an
                        // `SdkError::ServiceError`.
                        _ => (Err(SdkError::ServiceError(err)), String::new()),
                    }
                }
                // Any other kind of error is returned in a tuple containing the
                // error and an empty string.
                _ => (Err(err), String::new()),
            },
        };

    warn!("last modified: {last_modified:?}");
    assert_eq!(
        e_tag_1, e_tag_2,
        "PutObject and second HeadObject had different eTags"
    );

    println!("Second value of last modified: {last_modified:?}");
    println!("Second tag: {}", e_tag_2);

    // Clean up by deleting the object and the bucket.
    client
        .delete_object()
        .bucket(bucket_name.as_str())
        .key(KEY)
        .send()
        .await?;

    client
        .delete_bucket()
        .bucket(bucket_name.as_str())
        .send()
        .await?;

    Ok(())
```

```
}
```

- Pour plus de détails sur l'API, voir GetObjectla section de référence de l'API AWS SDK for Rust.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Commencez à utiliser les buckets et les objets Amazon S3 à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment :

- créer un compartiment et y charger un fichier ;
- télécharger un objet à partir d'un compartiment ;
- copier un objet dans le sous-dossier d'un compartiment ;
- répertorier les objets d'un compartiment ;
- supprimer le compartiment et tous les objets qui y figurent.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
public class S3_Basics
{
    public static async Task Main()
    {
        // Create an Amazon S3 client object. The constructor uses the
        // default user installed on the system. To work with Amazon S3
        // features in a different AWS Region, pass the AWS Region as a
```

```
            // parameter to the client constructor.
            IAmazonS3 client = new AmazonS3Client();
            string bucketName = string.Empty;
            string filePath = string.Empty;
            string keyName = string.Empty;

            var sepBar = new string('-', Console.WindowWidth);

            Console.WriteLine(sepBar);
            Console.WriteLine("Amazon Simple Storage Service (Amazon S3) basic");
            Console.WriteLine("procedures. This application will:");
            Console.WriteLine("\n\t1. Create a bucket");
            Console.WriteLine("\n\t2. Upload an object to the new bucket");
            Console.WriteLine("\n\t3. Copy the uploaded object to a folder in the
 bucket");
            Console.WriteLine("\n\t4. List the items in the new bucket");
            Console.WriteLine("\n\t5. Delete all the items in the bucket");
            Console.WriteLine("\n\t6. Delete the bucket");
            Console.WriteLine(sepBar);

            // Create a bucket.
            Console.WriteLine($"\n{sepBar}");
            Console.WriteLine("\nCreate a new Amazon S3 bucket.\n");
            Console.WriteLine(sepBar);

            Console.Write("Please enter a name for the new bucket: ");
            bucketName = Console.ReadLine();

            var success = await S3Bucket.CreateBucketAsync(client, bucketName);
            if (success)
            {
                Console.WriteLine($"Successfully created bucket: {bucketName}.
\n");
            }
            else
            {
                Console.WriteLine($"Could not create bucket: {bucketName}.\n");
            }

            Console.WriteLine(sepBar);
            Console.WriteLine("Upload a file to the new bucket.");
            Console.WriteLine(sepBar);

            // Get the local path and filename for the file to upload.
```

```
            while (string.IsNullOrEmpty(filePath))
            {
                Console.Write("Please enter the path and filename of the file to
upload: ");
                filePath = Console.ReadLine();

                // Confirm that the file exists on the local computer.
                if (!File.Exists(filePath))
                {
                    Console.WriteLine($"Couldn't find {filePath}. Try again.\n");
                    filePath = string.Empty;
                }
            }

            // Get the file name from the full path.
            keyName = Path.GetFileName(filePath);

            success = await S3Bucket.UploadFileAsync(client, bucketName, keyName,
filePath);

            if (success)
            {
                Console.WriteLine($"Successfully uploaded {keyName} from
{filePath} to {bucketName}.\n");
            }
            else
            {
                Console.WriteLine($"Could not upload {keyName}.\n");
            }

            // Set the file path to an empty string to avoid overwriting the
            // file we just uploaded to the bucket.
            filePath = string.Empty;

            // Now get a new location where we can save the file.
            while (string.IsNullOrEmpty(filePath))
            {
                // First get the path to which the file will be downloaded.
                Console.Write("Please enter the path where the file will be
downloaded: ");
                filePath = Console.ReadLine();

                // Confirm that the file exists on the local computer.
                if (File.Exists($"{filePath}\\{keyName}"))
```

```
                {
                    Console.WriteLine($"Sorry, the file already exists in that
location.\n");
                    filePath = string.Empty;
                }
            }

            // Download an object from a bucket.
            success = await S3Bucket.DownloadObjectFromBucketAsync(client,
bucketName, keyName, filePath);

            if (success)
            {
                Console.WriteLine($"Successfully downloaded {keyName}.\n");
            }
            else
            {
                Console.WriteLine($"Sorry, could not download {keyName}.\n");
            }

            // Copy the object to a different folder in the bucket.
            string folderName = string.Empty;

            while (string.IsNullOrEmpty(folderName))
            {
                Console.Write("Please enter the name of the folder to copy your
object to: ");
                folderName = Console.ReadLine();
            }

            while (string.IsNullOrEmpty(keyName))
            {
                // Get the name to give to the object once uploaded.
                Console.Write("Enter the name of the object to copy: ");
                keyName = Console.ReadLine();
            }

            await S3Bucket.CopyObjectInBucketAsync(client, bucketName, keyName,
folderName);

            // List the objects in the bucket.
            await S3Bucket.ListBucketContentsAsync(client, bucketName);

            // Delete the contents of the bucket.
```

```
            await S3Bucket.DeleteBucketContentsAsync(client, bucketName);

            // Deleting the bucket too quickly after deleting its contents will
            // cause an error that the bucket isn't empty. So...
            Console.WriteLine("Press <Enter> when you are ready to delete the
bucket.");
            _ = Console.ReadLine();

            // Delete the bucket.
            await S3Bucket.DeleteBucketAsync(client, bucketName);
        }
    }
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Bash

AWS CLI avec le script Bash

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
###############################################################################
# function s3_getting_started
#
```

```
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function s3_getting_started() {
  {
    if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
      cd bucket-lifecycle-operations || exit

      source ./bucket_operations.sh
      cd ..
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon S3 getting started demo."
  echo_repeat "*" 88

  local bucket_name
  bucket_name=$(generate_random_name "doc-example-bucket")

  local region_code
  region_code=$(aws configure get region)

  if create_bucket -b "$bucket_name" -r "$region_code"; then
    echo "Created demo bucket named $bucket_name"
  else
    errecho "The bucket failed to create. This demo will exit."
    return 1
  fi

  local file_name
  while [ -z "$file_name" ]; do
    echo -n "Enter a file you want to upload to your bucket: "
    get_input
    file_name=$get_input_result

    if [ ! -f "$file_name" ]; then
      echo "Could not find file $file_name. Are you sure it exists?"
      file_name=""
    fi
  done
```

```
 local key
 key="$(basename "$file_name")"

 local result=0
 if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
   echo "Uploaded file $file_name into bucket $bucket_name with key $key."
 else
   result=1
 fi

 local destination_file
 destination_file="$file_name.download"
 if yes_no_input "Would you like to download $key to the file $destination_file?
(y/n) "; then
   if download_object_from_bucket "$bucket_name" "$destination_file" "$key";
then
     echo "Downloaded $key in the bucket $bucket_name to the file
$destination_file."
   else
     result=1
   fi
 fi

 if yes_no_input "Would you like to copy $key a new object key in your bucket?
(y/n) "; then
   local to_key
   to_key="demo/$key"
   if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
     echo "Copied $key in the bucket $bucket_name to the  $to_key."
   else
     result=1
   fi
 fi

 local bucket_items
 bucket_items=$(list_items_in_bucket "$bucket_name")

 # shellcheck disable=SC2181
 if [[ $? -ne 0 ]]; then
   result=1
 fi

 echo "Your bucket contains the following items."
```

```
  echo -e "Name\t\tSize"
  echo "$bucket_items"

  if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
 (y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
      echo "The following items were deleted from the bucket $bucket_name"
      echo "$bucket_items"
    else
      result=1
    fi

    if delete_bucket "$bucket_name"; then
      echo "Deleted the bucket $bucket_name"
    else
      result=1
    fi
  fi

  return $result
}
```

Les fonctions Amazon S3 utilisées dans ce scénario.

```
###############################################################################
# function create-bucket
#
# This function creates the specified bucket in the specified AWS Region, unless
# it already exists.
#
# Parameters:
#       -b bucket_name  -- The name of the bucket to create.
#       -r region_code  -- The code for an AWS Region in which to
#                          create the bucket.
#
# Returns:
#       The URL of the bucket that was created.
#     And:
#       0 - If successful.
#       1 - If it fails.
```

```
########################################################################################
function create_bucket() {
  local bucket_name region_code response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally
 unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
```

```
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3
```

```
  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}
```

```
##############################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}

##############################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
```

```
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}

###############################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
```

```
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n
$response"
    return 1
  fi
}

################################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la Référence des commandes AWS CLI .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
#include <iostream>
#include <aws/core/Aws.h>
#include <aws/s3/S3Client.h>
#include <aws/s3/model/CopyObjectRequest.h>
#include <aws/s3/model/CreateBucketRequest.h>
#include <aws/s3/model/DeleteBucketRequest.h>
#include <aws/s3/model/DeleteObjectRequest.h>
#include <aws/s3/model/GetObjectRequest.h>
#include <aws/s3/model/ListObjectsV2Request.h>
#include <aws/s3/model/PutObjectRequest.h>
#include <aws/s3/model/BucketLocationConstraint.h>
#include <aws/s3/model/CreateBucketConfiguration.h>
#include <aws/core/utils/UUID.h>
#include <aws/core/utils/StringUtils.h>
#include <aws/core/utils/memory/stl/AWSAllocator.h>
#include <fstream>
```

```
#include "s3_examples.h"

namespace AwsDoc {
    namespace S3 {

        //! Delete an S3 bucket.
        /*!
          \param bucketName: The S3 bucket's name.
          \param client: An S3 client.
          \return bool: Function succeeded.
        */
        static bool
        deleteBucket(const Aws::String &bucketName, Aws::S3::S3Client &client);

        //! Delete an object in an S3 bucket.
        /*!
          \param bucketName: The S3 bucket's name.
          \param key: The key for the object in the S3 bucket.
          \param client: An S3 client.
          \return bool: Function succeeded.
         */
        static bool
        deleteObjectFromBucket(const Aws::String &bucketName, const Aws::String
 &key,
                               Aws::S3::S3Client &client);
    }
}


//! Scenario to create, copy, and delete S3 buckets and objects.
/*!
  \param uploadFilePath: Path to file to upload to an Amazon S3 bucket.
  \param saveFilePath: Path for saving a downloaded S3 object.
  \param clientConfig: Aws client configuration.
  \return bool: Function succeeded.
 */
bool AwsDoc::S3::S3_GettingStartedScenario(const Aws::String &uploadFilePath,
                                           const Aws::String &saveFilePath,
                                           const Aws::Client::ClientConfiguration
 &clientConfig) {

    Aws::S3::S3Client client(clientConfig);

    // Create a unique bucket name which is only temporary and will be deleted.
```

```
   // Format: "doc-example-bucket-" + lowercase UUID.
   Aws::String uuid = Aws::Utils::UUID::RandomUUID();
   Aws::String bucketName = "doc-example-bucket-" +
                            Aws::Utils::StringUtils::ToLower(uuid.c_str());

   // 1. Create a bucket.
   {
       Aws::S3::Model::CreateBucketRequest request;
       request.SetBucket(bucketName);

       if (clientConfig.region != Aws::Region::US_EAST_1) {
           Aws::S3::Model::CreateBucketConfiguration createBucketConfiguration;
           createBucketConfiguration.WithLocationConstraint(

Aws::S3::Model::BucketLocationConstraintMapper::GetBucketLocationConstraintForName(
                           clientConfig.region));
           request.WithCreateBucketConfiguration(createBucketConfiguration);
       }

       Aws::S3::Model::CreateBucketOutcome outcome =
client.CreateBucket(request);

       if (!outcome.IsSuccess()) {
           const Aws::S3::S3Error &err = outcome.GetError();
           std::cerr << "Error: createBucket: " <<
                     err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
           return false;
       } else {
           std::cout << "Created the bucket, '" << bucketName <<
                     "', in the region, '" << clientConfig.region << "'." <<
std::endl;
       }
   }

   // 2. Upload a local file to the bucket.
   Aws::String key = "key-for-test";
   {
       Aws::S3::Model::PutObjectRequest request;
       request.SetBucket(bucketName);
       request.SetKey(key);

       std::shared_ptr<Aws::FStream> input_data =
               Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
```

```
                                                uploadFilePath,
                                                std::ios_base::in |
                                                std::ios_base::binary);

        if (!input_data->is_open()) {
            std::cerr << "Error: unable to open file, '" << uploadFilePath <<
"'."
                      << std::endl;
            AwsDoc::S3::deleteBucket(bucketName, client);
            return false;
        }

        request.SetBody(input_data);

        Aws::S3::Model::PutObjectOutcome outcome =
                client.PutObject(request);

        if (!outcome.IsSuccess()) {
            std::cerr << "Error: putObject: " <<
                      outcome.GetError().GetMessage() << std::endl;
            AwsDoc::S3::deleteObjectFromBucket(bucketName, key, client);
            AwsDoc::S3::deleteBucket(bucketName, client);
            return false;
        } else {
            std::cout << "Added the object with the key, '" << key
                      << "', to the bucket, '"
                      << bucketName << "'." << std::endl;
        }
    }

    // 3. Download the object to a local file.
    {
        Aws::S3::Model::GetObjectRequest request;
        request.SetBucket(bucketName);
        request.SetKey(key);

        Aws::S3::Model::GetObjectOutcome outcome =
                client.GetObject(request);

        if (!outcome.IsSuccess()) {
            const Aws::S3::S3Error &err = outcome.GetError();
            std::cerr << "Error: getObject: " <<
                      err.GetExceptionName() << ": " << err.GetMessage() <<
std::endl;
```

```
        } else {
            std::cout << "Downloaded the object with the key, '" << key
                      << "', in the bucket, '"
                      << bucketName << "'." << std::endl;

            Aws::IOStream &ioStream = outcome.GetResultWithOwnership().
                    GetBody();
            Aws::OFStream outStream(saveFilePath,
                                    std::ios_base::out | std::ios_base::binary);
            if (!outStream.is_open()) {
                std::cout << "Error: unable to open file, '" << saveFilePath <<
"'."
                          << std::endl;
            } else {
                outStream << ioStream.rdbuf();
                std::cout << "Wrote the downloaded object to the file '"
                          << saveFilePath << "'." << std::endl;
            }
        }
    }

    // 4. Copy the object to a different "folder" in the bucket.
    Aws::String copiedToKey = "test-folder/" + key;
    {
        Aws::S3::Model::CopyObjectRequest request;
        request.WithBucket(bucketName)
                .WithKey(copiedToKey)
                .WithCopySource(bucketName + "/" + key);

        Aws::S3::Model::CopyObjectOutcome outcome =
                client.CopyObject(request);
        if (!outcome.IsSuccess()) {
            std::cerr << "Error: copyObject: " <<
                      outcome.GetError().GetMessage() << std::endl;
        } else {
            std::cout << "Copied the object with the key, '" << key
                      << "', to the key, '" << copiedToKey
                      << ", in the bucket, '" << bucketName << "'." << std::endl;
        }
    }

    // 5. List objects in the bucket.
    {
        Aws::S3::Model::ListObjectsV2Request request;
```

```
        request.WithBucket(bucketName);

        Aws::String continuationToken;
        Aws::Vector<Aws::S3::Model::Object> allObjects;

        do {
            if (!continuationToken.empty()) {
                request.SetContinuationToken(continuationToken);
            }
            Aws::S3::Model::ListObjectsV2Outcome outcome = client.ListObjectsV2(
                    request);

            if (!outcome.IsSuccess()) {
                std::cerr << "Error: ListObjects: " <<
                          outcome.GetError().GetMessage() << std::endl;
                break;
            } else {
                Aws::Vector<Aws::S3::Model::Object> objects =
                        outcome.GetResult().GetContents();
                allObjects.insert(allObjects.end(), objects.begin(),
 objects.end());
                continuationToken = outcome.GetResult().GetContinuationToken();
            }
        } while (!continuationToken.empty());

        std::cout << allObjects.size() << " objects in the bucket, '" <<
 bucketName
                  << "':" << std::endl;

        for (Aws::S3::Model::Object &object: allObjects) {
            std::cout << "     '" << object.GetKey() << "'" << std::endl;
        }
    }

    // 6. Delete all objects in the bucket.
    // All objects in the bucket must be deleted before deleting the bucket.
    AwsDoc::S3::deleteObjectFromBucket(bucketName, copiedToKey, client);
    AwsDoc::S3::deleteObjectFromBucket(bucketName, key, client);

    // 7. Delete the bucket.
    return AwsDoc::S3::deleteBucket(bucketName, client);
}

bool AwsDoc::S3::deleteObjectFromBucket(const Aws::String &bucketName,
```

```
                                        const Aws::String &key,
                                        Aws::S3::S3Client &client) {
    Aws::S3::Model::DeleteObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(key);

    Aws::S3::Model::DeleteObjectOutcome outcome =
            client.DeleteObject(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: deleteObject: " <<
                  outcome.GetError().GetMessage() << std::endl;
    } else {
        std::cout << "Deleted the object with the key, '" << key
                  << "', from the bucket, '"
                  << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

bool
AwsDoc::S3::deleteBucket(const Aws::String &bucketName, Aws::S3::S3Client
 &client) {
    Aws::S3::Model::DeleteBucketRequest request;
    request.SetBucket(bucketName);

    Aws::S3::Model::DeleteBucketOutcome outcome =
            client.DeleteBucket(request);

    if (!outcome.IsSuccess()) {
        const Aws::S3::S3Error &err = outcome.GetError();
        std::cerr << "Error: deleteBucket: " <<
                  err.GetExceptionName() << ": " << err.GetMessage() <<
 std::endl;
    } else {
        std::cout << "Deleted the bucket, '" << bucketName << "'." << std::endl;
    }
    return outcome.IsSuccess();
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for C++ .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Définissez une structure qui enveloppe les actions de compartiment et d'objet utilisées par le scénario.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// ListBuckets lists the buckets in the current account.
func (basics BucketBasics) ListBuckets() ([]types.Bucket, error) {
```

```
 result, err := basics.S3Client.ListBuckets(context.TODO(),
 &s3.ListBucketsInput{})
 var buckets []types.Bucket
 if err != nil {
  log.Printf("Couldn't list buckets for your account. Here's why: %v\n", err)
 } else {
  buckets = result.Buckets
 }
 return buckets, err
}



// BucketExists checks whether a bucket exists in the current account.
func (basics BucketBasics) BucketExists(bucketName string) (bool, error) {
 _, err := basics.S3Client.HeadBucket(context.TODO(), &s3.HeadBucketInput{
  Bucket: aws.String(bucketName),
 })
 exists := true
 if err != nil {
  var apiError smithy.APIError
  if errors.As(err, &apiError) {
   switch apiError.(type) {
   case *types.NotFound:
    log.Printf("Bucket %v is available.\n", bucketName)
    exists = false
    err = nil
   default:
    log.Printf("Either you don't have access to bucket %v or another error
 occurred. "+
     "Here's what happened: %v\n", bucketName, err)
   }
  }
 } else {
  log.Printf("Bucket %v exists and you already own it.", bucketName)
 }

 return exists, err
}



// CreateBucket creates a bucket with the specified name in the specified Region.
func (basics BucketBasics) CreateBucket(name string, region string) error {
```

```
 _, err := basics.S3Client.CreateBucket(context.TODO(), &s3.CreateBucketInput{
  Bucket: aws.String(name),
  CreateBucketConfiguration: &types.CreateBucketConfiguration{
   LocationConstraint: types.BucketLocationConstraint(region),
  },
 })
 if err != nil {
  log.Printf("Couldn't create bucket %v in Region %v. Here's why: %v\n",
   name, region, err)
 }
 return err
}



// UploadFile reads from a file and puts the data into an object in a bucket.
func (basics BucketBasics) UploadFile(bucketName string, objectKey string,
 fileName string) error {
 file, err := os.Open(fileName)
 if err != nil {
  log.Printf("Couldn't open file %v to upload. Here's why: %v\n", fileName, err)
 } else {
  defer file.Close()
  _, err = basics.S3Client.PutObject(context.TODO(), &s3.PutObjectInput{
   Bucket: aws.String(bucketName),
   Key:    aws.String(objectKey),
   Body:   file,
  })
  if err != nil {
   log.Printf("Couldn't upload file %v to %v:%v. Here's why: %v\n",
    fileName, bucketName, objectKey, err)
  }
 }
 return err
}



// UploadLargeObject uses an upload manager to upload data to an object in a
 bucket.
// The upload manager breaks large data into parts and uploads the parts
 concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
 largeObject []byte) error {
```

```
 largeBuffer := bytes.NewReader(largeObject)
 var partMiBs int64 = 10
 uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
  u.PartSize = partMiBs * 1024 * 1024
 })
 _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
  Body:   largeBuffer,
 })
 if err != nil {
  log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }

 return err
}



// DownloadFile gets an object from a bucket and stores it in a local file.
func (basics BucketBasics) DownloadFile(bucketName string, objectKey string,
 fileName string) error {
 result, err := basics.S3Client.GetObject(context.TODO(), &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't get object %v:%v. Here's why: %v\n", bucketName,
 objectKey, err)
  return err
 }
 defer result.Body.Close()
 file, err := os.Create(fileName)
 if err != nil {
  log.Printf("Couldn't create file %v. Here's why: %v\n", fileName, err)
  return err
 }
 defer file.Close()
 body, err := io.ReadAll(result.Body)
 if err != nil {
  log.Printf("Couldn't read object body from %v. Here's why: %v\n", objectKey,
 err)
 }
```

```
 _, err = file.Write(body)
 return err
}



// DownloadLargeObject uses a download manager to download an object from a
 bucket.
// The download manager gets the data in parts and writes them to a buffer until
 all of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
 string) ([]byte, error) {
 var partMiBs int64 = 10
 downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
 {
  d.PartSize = partMiBs * 1024 * 1024
 })
 buffer := manager.NewWriteAtBuffer([]byte{})
 _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return buffer.Bytes(), err
}



// CopyToFolder copies an object in a bucket to a subfolder in the same bucket.
func (basics BucketBasics) CopyToFolder(bucketName string, objectKey string,
 folderName string) error {
 _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
  Bucket:     aws.String(bucketName),
  CopySource: aws.String(fmt.Sprintf("%v/%v", bucketName, objectKey)),
  Key:        aws.String(fmt.Sprintf("%v/%v", folderName, objectKey)),
 })
 if err != nil {
  log.Printf("Couldn't copy object from %v:%v to %v:%v/%v. Here's why: %v\n",
   bucketName, objectKey, bucketName, folderName, objectKey, err)
 }
```

```
 return err
}



// CopyToBucket copies an object in a bucket to another bucket.
func (basics BucketBasics) CopyToBucket(sourceBucket string, destinationBucket
 string, objectKey string) error {
 _, err := basics.S3Client.CopyObject(context.TODO(), &s3.CopyObjectInput{
  Bucket:     aws.String(destinationBucket),
  CopySource: aws.String(fmt.Sprintf("%v/%v", sourceBucket, objectKey)),
  Key:        aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't copy object from %v:%v to %v:%v. Here's why: %v\n",
   sourceBucket, objectKey, destinationBucket, objectKey, err)
 }
 return err
}



// ListObjects lists the objects in a bucket.
func (basics BucketBasics) ListObjects(bucketName string) ([]types.Object, error)
 {
 result, err := basics.S3Client.ListObjectsV2(context.TODO(),
 &s3.ListObjectsV2Input{
  Bucket: aws.String(bucketName),
 })
 var contents []types.Object
 if err != nil {
  log.Printf("Couldn't list objects in bucket %v. Here's why: %v\n", bucketName,
 err)
 } else {
  contents = result.Contents
 }
 return contents, err
}



// DeleteObjects deletes a list of objects from a bucket.
func (basics BucketBasics) DeleteObjects(bucketName string, objectKeys []string)
 error {
```

```
 var objectIds []types.ObjectIdentifier
 for _, key := range objectKeys {
  objectIds = append(objectIds, types.ObjectIdentifier{Key: aws.String(key)})
 }
 output, err := basics.S3Client.DeleteObjects(context.TODO(),
 &s3.DeleteObjectsInput{
  Bucket: aws.String(bucketName),
  Delete: &types.Delete{Objects: objectIds},
 })
 if err != nil {
  log.Printf("Couldn't delete objects from bucket %v. Here's why: %v\n",
 bucketName, err)
 } else {
  log.Printf("Deleted %v objects.\n", len(output.Deleted))
 }
 return err
}



// DeleteBucket deletes a bucket. The bucket must be empty or an error is
 returned.
func (basics BucketBasics) DeleteBucket(bucketName string) error {
 _, err := basics.S3Client.DeleteBucket(context.TODO(), &s3.DeleteBucketInput{
  Bucket: aws.String(bucketName)})
 if err != nil {
  log.Printf("Couldn't delete bucket %v. Here's why: %v\n", bucketName, err)
 }
 return err
}
```

Exécutez un scénario interactif qui vous montre comment utiliser des compartiments et des objets S3.

```


// RunGetStartedScenario is an interactive example that shows you how to use
 Amazon
// Simple Storage Service (Amazon S3) to create an S3 bucket and use it to store
 objects.
//
// 1. Create a bucket.
```

```
// 2. Upload a local file to the bucket.
// 3. Upload a large object to the bucket by using an upload manager.
// 4. Download an object to a local file.
// 5. Download a large object by using a download manager.
// 6. Copy an object to a different folder in the bucket.
// 7. List objects in the bucket.
// 8. Delete all objects in the bucket.
// 9. Delete the bucket.
//
// This example creates an Amazon S3 service client from the specified sdkConfig
 so that
// you can replace it with a mocked or stubbed config for unit testing.
//
// It uses a questioner from the `demotools` package to get input during the
 example.
// This package can be found in the ..\..\demotools folder of this repo.
func RunGetStartedScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner) {
 defer func() {
  if r := recover(); r != nil {
   fmt.Println("Something went wrong with the demo.\n", r)
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the Amazon S3 getting started demo.")
 log.Println(strings.Repeat("-", 88))

 s3Client := s3.NewFromConfig(sdkConfig)
 bucketBasics := actions.BucketBasics{S3Client: s3Client}

 count := 10
 log.Printf("Let's list up to %v buckets for your account:", count)
 buckets, err := bucketBasics.ListBuckets()
 if err != nil {
  panic(err)
 }
 if len(buckets) == 0 {
  log.Println("You don't have any buckets!")
 } else {
  if count > len(buckets) {
   count = len(buckets)
  }
  for _, bucket := range buckets[:count] {
```

```
  log.Printf("\t%v\n", *bucket.Name)
 }
}

bucketName := questioner.Ask("Let's create a bucket. Enter a name for your
bucket:",
 demotools.NotEmpty{})
bucketExists, err := bucketBasics.BucketExists(bucketName)
if err != nil {
 panic(err)
}
if !bucketExists {
 err = bucketBasics.CreateBucket(bucketName, sdkConfig.Region)
 if err != nil {
  panic(err)
 } else {
  log.Println("Bucket created.")
 }
}
log.Println(strings.Repeat("-", 88))

fmt.Println("Let's upload a file to your bucket.")
smallFile := questioner.Ask("Enter the path to a file you want to upload:",
 demotools.NotEmpty{})
const smallKey = "doc-example-key"
err = bucketBasics.UploadFile(bucketName, smallKey, smallFile)
if err != nil {
 panic(err)
}
log.Printf("Uploaded %v as %v.\n", smallFile, smallKey)
log.Println(strings.Repeat("-", 88))

mibs := 30
log.Printf("Let's create a slice of %v MiB of random bytes and upload it to your
bucket. ", mibs)
questioner.Ask("Press Enter when you're ready.")
largeBytes := make([]byte, 1024*1024*mibs)
rand.Seed(time.Now().Unix())
rand.Read(largeBytes)
largeKey := "doc-example-large"
log.Println("Uploading...")
err = bucketBasics.UploadLargeObject(bucketName, largeKey, largeBytes)
if err != nil {
 panic(err)
```

```
}
log.Printf("Uploaded %v MiB object as %v", mibs, largeKey)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download %v to a file.", smallKey)
downloadFileName := questioner.Ask("Enter a name for the downloaded file:",
demotools.NotEmpty{})
err = bucketBasics.DownloadFile(bucketName, smallKey, downloadFileName)
if err != nil {
 panic(err)
}
log.Printf("File %v downloaded.", downloadFileName)
log.Println(strings.Repeat("-", 88))

log.Printf("Let's download the %v MiB object.", mibs)
questioner.Ask("Press Enter when you're ready.")
log.Println("Downloading...")
largeDownload, err := bucketBasics.DownloadLargeObject(bucketName, largeKey)
if err != nil {
 panic(err)
}
log.Printf("Downloaded %v bytes.", len(largeDownload))
log.Println(strings.Repeat("-", 88))

log.Printf("Let's copy %v to a folder in the same bucket.", smallKey)
folderName := questioner.Ask("Enter a folder name: ", demotools.NotEmpty{})
err = bucketBasics.CopyToFolder(bucketName, smallKey, folderName)
if err != nil {
 panic(err)
}
log.Printf("Copied %v to %v/%v.\n", smallKey, folderName, smallKey)
log.Println(strings.Repeat("-", 88))

log.Println("Let's list the objects in your bucket.")
questioner.Ask("Press Enter when you're ready.")
objects, err := bucketBasics.ListObjects(bucketName)
if err != nil {
 panic(err)
}
log.Printf("Found %v objects.\n", len(objects))
var objKeys []string
for _, object := range objects {
 objKeys = append(objKeys, *object.Key)
 log.Printf("\t%v\n", *object.Key)
```

```
 }
 log.Println(strings.Repeat("-", 88))

 if questioner.AskBool("Do you want to delete your bucket and all of its "+
  "contents? (y/n)", "y") {
  log.Println("Deleting objects.")
  err = bucketBasics.DeleteObjects(bucketName, objKeys)
  if err != nil {
   panic(err)
  }
  log.Println("Deleting bucket.")
  err = bucketBasics.DeleteBucket(bucketName)
  if err != nil {
   panic(err)
  }
  log.Printf("Deleting downloaded file %v.\n", downloadFileName)
  err = os.Remove(downloadFileName)
  if err != nil {
   panic(err)
  }
 } else {
  log.Println("Okay. Don't forget to delete objects from your bucket to avoid
 charges.")
 }
 log.Println(strings.Repeat("-", 88))

 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Go .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2

- PutObject

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This Java code example performs the following tasks:
 *
 * 1. Creates an Amazon S3 bucket.
 * 2. Uploads an object to the bucket.
 * 3. Downloads the object to another local file.
 * 4. Uploads an object using multipart upload.
 * 5. List all objects located in the Amazon S3 bucket.
 * 6. Copies the object to another Amazon S3 bucket.
 * 7. Deletes the object from the Amazon S3 bucket.
 * 8. Deletes the Amazon S3 bucket.
 */

public class S3Scenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
```

```
                    <bucketName> <key> <objectPath> <savePath> <toBucket>

                Where:
                    bucketName - The Amazon S3 bucket to create.
                    key - The key to use.
                    objectPath - The path where the file is located (for example,
C:/AWS/book2.pdf).
                    savePath - The path where the file is saved after it's
downloaded (for example, C:/AWS/book2.pdf).
                    toBucket - An Amazon S3 bucket to where an object is copied
to (for example, C:/AWS/book2.pdf).\s
                    """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String bucketName = args[0];
        String key = args[1];
        String objectPath = args[2];
        String savePath = args[3];
        String toBucket = args[4];
        Region region = Region.US_EAST_1;
        S3Client s3 = S3Client.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon S3 example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("1. Create an Amazon S3 bucket.");
        createBucket(s3, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Update a local file to the Amazon S3 bucket.");
        uploadLocalFile(s3, bucketName, key, objectPath);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Download the object to another local file.");
```

```
        getObjectBytes(s3, bucketName, key, savePath);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Perform a multipart upload.");
        String multipartKey = "multiPartKey";
        multipartUpload(s3, toBucket, multipartKey);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. List all objects located in the Amazon S3
bucket.");
        listAllObjects(s3, bucketName);
        anotherListExample(s3, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Copy the object to another Amazon S3 bucket.");
        copyBucketObject(s3, bucketName, key, toBucket);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Delete the object from the Amazon S3 bucket.");
        deleteObjectFromBucket(s3, bucketName, key);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Delete the Amazon S3 bucket.");
        deleteBucket(s3, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("All Amazon S3 operations were successfully
performed");
        System.out.println(DASHES);
        s3.close();
    }

    // Create a bucket by using a S3Waiter object.
    public static void createBucket(S3Client s3Client, String bucketName) {
        try {
            S3Waiter s3Waiter = s3Client.waiter();
            CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
                    .bucket(bucketName)
```

```
                    .build();

            s3Client.createBucket(bucketRequest);
            HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            // Wait until the bucket is created and print out the response.
            WaiterResponse<HeadBucketResponse> waiterResponse =
s3Waiter.waitUntilBucketExists(bucketRequestWait);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println(bucketName + " is ready");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteBucket(S3Client client, String bucket) {
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucket)
                .build();

        client.deleteBucket(deleteBucketRequest);
        System.out.println(bucket + " was deleted.");
    }

    /**
     * Upload an object in parts.
     */
    public static void multipartUpload(S3Client s3, String bucketName, String
key) {
        int mB = 1024 * 1024;
        // First create a multipart upload and get the upload id.
        CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();

        CreateMultipartUploadResponse response =
s3.createMultipartUpload(createMultipartUploadRequest);
        String uploadId = response.uploadId();
```

```
        System.out.println(uploadId);

        // Upload all the different parts of the object.
        UploadPartRequest uploadPartRequest1 = UploadPartRequest.builder()
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)
                .partNumber(1).build();

        String etag1 = s3.uploadPart(uploadPartRequest1,
RequestBody.fromByteBuffer(getRandomByteBuffer(5 * mB)))
                .eTag();
        CompletedPart part1 =
CompletedPart.builder().partNumber(1).eTag(etag1).build();

        UploadPartRequest uploadPartRequest2 =
UploadPartRequest.builder().bucket(bucketName).key(key)
                .uploadId(uploadId)
                .partNumber(2).build();
        String etag2 = s3.uploadPart(uploadPartRequest2,
RequestBody.fromByteBuffer(getRandomByteBuffer(3 * mB)))
                .eTag();
        CompletedPart part2 =
CompletedPart.builder().partNumber(2).eTag(etag2).build();

        // Call completeMultipartUpload operation to tell S3 to merge all
uploaded
        // parts and finish the multipart operation.
        CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder()
                .parts(part1, part2)
                .build();

        CompleteMultipartUploadRequest completeMultipartUploadRequest =
CompleteMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)
                .multipartUpload(completedMultipartUpload)
                .build();

        s3.completeMultipartUpload(completeMultipartUploadRequest);
    }
```

```
   private static ByteBuffer getRandomByteBuffer(int size) {
       byte[] b = new byte[size];
       new Random().nextBytes(b);
       return ByteBuffer.wrap(b);
   }

   public static void getObjectBytes(S3Client s3, String bucketName, String
keyName, String path) {
       try {
           GetObjectRequest objectRequest = GetObjectRequest
                   .builder()
                   .key(keyName)
                   .bucket(bucketName)
                   .build();

           ResponseBytes<GetObjectResponse> objectBytes =
s3.getObjectAsBytes(objectRequest);
           byte[] data = objectBytes.asByteArray();

           // Write the data to a local file.
           File myFile = new File(path);
           OutputStream os = new FileOutputStream(myFile);
           os.write(data);
           System.out.println("Successfully obtained bytes from an S3 object");
           os.close();

       } catch (IOException ex) {
           ex.printStackTrace();
       } catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }

   public static void uploadLocalFile(S3Client s3, String bucketName, String
key, String objectPath) {
       PutObjectRequest objectRequest = PutObjectRequest.builder()
               .bucket(bucketName)
               .key(key)
               .build();

       s3.putObject(objectRequest, RequestBody.fromFile(new File(objectPath)));
   }
```

```
   public static void listAllObjects(S3Client s3, String bucketName) {
       ListObjectsV2Request listObjectsReqManual =
ListObjectsV2Request.builder()
               .bucket(bucketName)
               .maxKeys(1)
               .build();

       boolean done = false;
       while (!done) {
           ListObjectsV2Response listObjResponse =
s3.listObjectsV2(listObjectsReqManual);
           for (S3Object content : listObjResponse.contents()) {
               System.out.println(content.key());
           }

           if (listObjResponse.nextContinuationToken() == null) {
               done = true;
           }

           listObjectsReqManual = listObjectsReqManual.toBuilder()
                   .continuationToken(listObjResponse.nextContinuationToken())
                   .build();
       }
   }

   public static void anotherListExample(S3Client s3, String bucketName) {
       ListObjectsV2Request listReq = ListObjectsV2Request.builder()
               .bucket(bucketName)
               .maxKeys(1)
               .build();

       ListObjectsV2Iterable listRes = s3.listObjectsV2Paginator(listReq);

       // Process response pages.
       listRes.stream()
               .flatMap(r -> r.contents().stream())
               .forEach(content -> System.out.println(" Key: " + content.key() +
" size = " + content.size()));

       // Helper method to work with paginated collection of items directly.
       listRes.contents().stream()
               .forEach(content -> System.out.println(" Key: " + content.key() +
" size = " + content.size()));
```

```
        for (S3Object content : listRes.contents()) {
            System.out.println(" Key: " + content.key() + " size = " +
content.size());
        }
    }

    public static void deleteObjectFromBucket(S3Client s3, String bucketName,
String key) {
        DeleteObjectRequest deleteObjectRequest = DeleteObjectRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();

        s3.deleteObject(deleteObjectRequest);
        System.out.println(key + " was deleted");
    }

    public static String copyBucketObject(S3Client s3, String fromBucket, String
objectKey, String toBucket) {
        String encodedUrl = null;
        try {
            encodedUrl = URLEncoder.encode(fromBucket + "/" + objectKey,
StandardCharsets.UTF_8.toString());
        } catch (UnsupportedEncodingException e) {
            System.out.println("URL could not be encoded: " + e.getMessage());
        }
        CopyObjectRequest copyReq = CopyObjectRequest.builder()
                .copySource(encodedUrl)
                .destinationBucket(toBucket)
                .destinationKey(objectKey)
                .build();

        try {
            CopyObjectResponse copyRes = s3.copyObject(copyReq);
            System.out.println("The " + objectKey + " was copied to " +
toBucket);
            return copyRes.copyObjectResult().toString();

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
```

```
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Tout d'abord, importez tous les modules nécessaires.

```
// Used to check if currently running file is this file.
import { fileURLToPath } from "url";
import { readdirSync, readFileSync, writeFileSync } from "fs";

// Local helper utils.
import { dirnameFromMetaUrl } from "@aws-doc-sdk-examples/lib/utils/util-fs.js";
import { Prompter } from "@aws-doc-sdk-examples/lib/prompter.js";
import { wrapText } from "@aws-doc-sdk-examples/lib/utils/util-string.js";

import {
  S3Client,
  CreateBucketCommand,
  PutObjectCommand,
  ListObjectsCommand,
```

```
  CopyObjectCommand,
  GetObjectCommand,
  DeleteObjectsCommand,
  DeleteBucketCommand,
} from "@aws-sdk/client-s3";
```

Les importations précédentes font référence à certains utilitaires d'annotations. Ces utilitaires sont locaux au GitHub référentiel lié au début de cette section. À titre de référence, consultez les implémentations suivantes de ces utilitaires.

```
export const dirnameFromMetaUrl = (metaUrl) =>
  fileURLToPath(new URL(".", metaUrl));

import { select, input, confirm, checkbox } from "@inquirer/prompts";

export class Prompter {
  /**
   * @param {{ message: string, choices: { name: string, value: string }[]}}
 options
   */
  select(options) {
    return select(options);
  }

  /**
   * @param {{ message: string }} options
   */
  input(options) {
    return input(options);
  }

  /**
   * @param {string} prompt
   */
  checkContinue = async (prompt = "") => {
    const prefix = prompt && prompt + " ";
    let ok = await this.confirm({
      message: `${prefix}Continue?`,
    });
    if (!ok) throw new Error("Exiting...");
  };
```

```
  /**
   * @param {{ message: string }} options
   */
  confirm(options) {
    return confirm(options);
  }

  /**
   * @param {{ message: string, choices: { name: string, value: string }[]}}
 options
   */
  checkbox(options) {
    return checkbox(options);
  }
}

export const wrapText = (text, char = "=") => {
  const rule = char.repeat(80);
  return `${rule}\n    ${text}\n${rule}\n`;
};
```

Les objets dans S3 sont stockés dans des « compartiments ». Définissons une fonction pour créer un nouveau compartiment.

```
export const createBucket = async () => {
  const bucketName = await prompter.input({
    message: "Enter a bucket name. Bucket names must be globally unique:",
  });
  const command = new CreateBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log("Bucket created successfully.\n");
  return bucketName;
};
```

Les compartiments contiennent des « objets ». Cette fonction charge le contenu d'un répertoire dans votre compartiment sous forme d'objets.

```
export const uploadFilesToBucket = async ({ bucketName, folderPath }) => {
  console.log(`Uploading files from ${folderPath}\n`);
  const keys = readdirSync(folderPath);
  const files = keys.map((key) => {
```

```
    const filePath = `${folderPath}/${key}`;
    const fileContent = readFileSync(filePath);
    return {
      Key: key,
      Body: fileContent,
    };
  });

  for (let file of files) {
    await s3Client.send(
      new PutObjectCommand({
        Bucket: bucketName,
        Body: file.Body,
        Key: file.Key,
      }),
    );
    console.log(`${file.Key} uploaded successfully.`);
  }
};
```

Après avoir chargé des objets, vérifiez qu'ils ont été chargés correctement. Tu peux l'utiliser ListObjects pour ça. Vous utiliserez la propriété « Key » (Clé), mais la réponse contient également d'autres propriétés utiles.

```
export const listFilesInBucket = async ({ bucketName }) => {
  const command = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(command);
  const contentsList = Contents.map((c) => ` • ${c.Key}`).join("\n");
  console.log("\nHere's a list of files in the bucket:");
  console.log(contentsList + "\n");
};
```

Il se peut que vous souhaitiez copier un objet d'un compartiment à un autre. Utilisez la CopyObject commande pour cela.

```
export const copyFileFromBucket = async ({ destinationBucket }) => {
  const proceed = await prompter.confirm({
    message: "Would you like to copy an object from another bucket?",
  });
```

```
  if (!proceed) {
    return;
  } else {
    const copy = async () => {
      try {
        const sourceBucket = await prompter.input({
          message: "Enter source bucket name:",
        });
        const sourceKey = await prompter.input({
          message: "Enter source key:",
        });
        const destinationKey = await prompter.input({
          message: "Enter destination key:",
        });

        const command = new CopyObjectCommand({
          Bucket: destinationBucket,
          CopySource: `${sourceBucket}/${sourceKey}`,
          Key: destinationKey,
        });
        await s3Client.send(command);
        await copyFileFromBucket({ destinationBucket });
      } catch (err) {
        console.error(`Copy error.`);
        console.error(err);
        const retryAnswer = await prompter.confirm({ message: "Try again?" });
        if (retryAnswer) {
          await copy();
        }
      }
    };
    await copy();
  }
};
```

Il n'existe aucune méthode de kit SDK pour obtenir plusieurs objets d'un compartiment. Vous allez plutôt créer une liste d'objets à charger et sur laquelle itérer.

```
export const downloadFilesFromBucket = async ({ bucketName }) => {
  const { Contents } = await s3Client.send(
    new ListObjectsCommand({ Bucket: bucketName }),
  );
```

```
  const path = await prompter.input({
    message: "Enter destination path for files:",
  });

  for (let content of Contents) {
    const obj = await s3Client.send(
      new GetObjectCommand({ Bucket: bucketName, Key: content.Key }),
    );
    writeFileSync(
      `${path}/${content.Key}`,
      await obj.Body.transformToByteArray(),
    );
  }
  console.log("Files downloaded successfully.\n");
};
```

Il est temps de nettoyer vos ressources. Un compartiment doit être vide avant de pouvoir être supprimé. Ces deux fonctions vident et suppriment le compartiment.

```
export const emptyBucket = async ({ bucketName }) => {
  const listObjectsCommand = new ListObjectsCommand({ Bucket: bucketName });
  const { Contents } = await s3Client.send(listObjectsCommand);
  const keys = Contents.map((c) => c.Key);

  const deleteObjectsCommand = new DeleteObjectsCommand({
    Bucket: bucketName,
    Delete: { Objects: keys.map((key) => ({ Key: key })) },
  });
  await s3Client.send(deleteObjectsCommand);
  console.log(`${bucketName} emptied successfully.\n`);
};

export const deleteBucket = async ({ bucketName }) => {
  const command = new DeleteBucketCommand({ Bucket: bucketName });
  await s3Client.send(command);
  console.log(`${bucketName} deleted successfully.\n`);
};
```

La fonction « main » regroupe tout. Si vous exécutez ce fichier directement, la fonction « main » sera appelée.

```
const main = async () => {
  const OBJECT_DIRECTORY = `${dirnameFromMetaUrl(
    import.meta.url,
  )}../../../../resources/sample_files/.sample_media`;

  try {
    console.log(wrapText("Welcome to the Amazon S3 getting started example."));
    console.log("Let's create a bucket.");
    const bucketName = await createBucket();
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("File upload."));
    console.log(
      "I have some default files ready to go. You can edit the source code to
 provide your own.",
    );
    await uploadFilesToBucket({
      bucketName,
      folderPath: OBJECT_DIRECTORY,
    });

    await listFilesInBucket({ bucketName });
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("Copy files."));
    await copyFileFromBucket({ destinationBucket: bucketName });
    await listFilesInBucket({ bucketName });
    await prompter.confirm({ message: continueMessage });

    console.log(wrapText("Download files."));
    await downloadFilesFromBucket({ bucketName });

    console.log(wrapText("Clean up."));
    await emptyBucket({ bucketName });
    await deleteBucket({ bucketName });
  } catch (err) {
    console.error(err);
  }
};
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for JavaScript .

- CopyObject
- CreateBucket
- DeleteBucket
- DeleteObjects
- GetObject
- ListObjectsV2
- PutObject

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <bucketName> <key> <objectPath> <savePath> <toBucket>

    Where:
        bucketName - The Amazon S3 bucket to create.
        key - The key to use.
        objectPath - The path where the file is located (for example, C:/AWS/
book2.pdf).
        savePath - The path where the file is saved after it's downloaded (for
 example, C:/AWS/book2.pdf).
        toBucket - An Amazon S3 bucket to where an object is copied to (for
 example, C:/AWS/book2.pdf).
        """

    if (args.size != 4) {
        println(usage)
        exitProcess(1)
    }
```

```
    val bucketName = args[0]
    val key = args[1]
    val objectPath = args[2]
    val savePath = args[3]
    val toBucket = args[4]

    // Create an Amazon S3 bucket.
    createBucket(bucketName)

    // Update a local file to the Amazon S3 bucket.
    putObject(bucketName, key, objectPath)

    // Download the object to another local file.
    getObjectFromMrap(bucketName, key, savePath)

    // List all objects located in the Amazon S3 bucket.
    listBucketObs(bucketName)

    // Copy the object to another Amazon S3 bucket
    copyBucketOb(bucketName, key, toBucket)

    // Delete the object from the Amazon S3 bucket.
    deleteBucketObs(bucketName, key)

    // Delete the Amazon S3 bucket.
    deleteBucket(bucketName)
    println("All Amazon S3 operations were successfully performed")
}

suspend fun createBucket(bucketName: String) {
    val request =
        CreateBucketRequest {
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        println("$bucketName is ready")
    }
}

suspend fun putObject(
    bucketName: String,
    objectKey: String,
```

```
    objectPath: String,
) {
    val metadataVal = mutableMapOf<String, String>()
    metadataVal["myVal"] = "test"

    val request =
        PutObjectRequest {
            bucket = bucketName
            key = objectKey
            metadata = metadataVal
            this.body = Paths.get(objectPath).asByteStream()
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        val response = s3.putObject(request)
        println("Tag information is ${response.eTag}")
    }
}

suspend fun getObjectFromMrap(
    bucketName: String,
    keyName: String,
    path: String,
) {
    val request =
        GetObjectRequest {
            key = keyName
            bucket = bucketName
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.getObject(request) { resp ->
            val myFile = File(path)
            resp.body?.writeToFile(myFile)
            println("Successfully read $keyName from $bucketName")
        }
    }
}

suspend fun listBucketObs(bucketName: String) {
    val request =
        ListObjectsRequest {
            bucket = bucketName
        }
```

```
    S3Client { region = "us-east-1" }.use { s3 ->

        val response = s3.listObjects(request)
        response.contents?.forEach { myObject ->
            println("The name of the key is ${myObject.key}")
            println("The owner is ${myObject.owner}")
        }
    }
}

suspend fun copyBucketOb(
    fromBucket: String,
    objectKey: String,
    toBucket: String,
) {
    var encodedUrl = ""
    try {
        encodedUrl = URLEncoder.encode("$fromBucket/$objectKey",
 StandardCharsets.UTF_8.toString())
    } catch (e: UnsupportedEncodingException) {
        println("URL could not be encoded: " + e.message)
    }

    val request =
        CopyObjectRequest {
            copySource = encodedUrl
            bucket = toBucket
            key = objectKey
        }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.copyObject(request)
    }
}

suspend fun deleteBucketObs(
    bucketName: String,
    objectName: String,
) {
    val objectId =
        ObjectIdentifier {
            key = objectName
        }
```

```
    val delOb =
        Delete {
            objects = listOf(objectId)
        }

    val request =
        DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteObjects(request)
        println("$objectName was deleted from $bucketName")
    }
}

suspend fun deleteBucket(bucketName: String?) {
    val request =
        DeleteBucketRequest {
            bucket = bucketName
        }
    S3Client { region = "us-east-1" }.use { s3 ->
        s3.deleteBucket(request)
        println("The $bucketName was successfully deleted!")
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Kotlin API reference.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
        echo("\n");
        echo("---------------------------------------\n");
        print("Welcome to the Amazon S3 getting started demo using PHP!\n");
        echo("---------------------------------------\n");

        $region = 'us-west-2';

        $this->s3client = new S3Client([
                'region' => $region,
        ]);
        /* Inline declaration example
        $s3client = new Aws\S3\S3Client(['region' => 'us-west-2']);
        */

        $this->bucketName = "doc-example-bucket-" . uniqid();

        try {
            $this->s3client->createBucket([
                'Bucket' => $this->bucketName,
                'CreateBucketConfiguration' => ['LocationConstraint' => $region],
            ]);
            echo "Created bucket named: $this->bucketName \n";
        } catch (Exception $exception) {
            echo "Failed to create bucket $this->bucketName with error: " .
$exception->getMessage();
            exit("Please fix error with bucket creation before continuing.");
        }

        $fileName = __DIR__ . "/local-file-" . uniqid();
        try {
            $this->s3client->putObject([
                'Bucket' => $this->bucketName,
```

```
                'Key' => $fileName,
                'SourceFile' => __DIR__ . '/testfile.txt'
            ]);
            echo "Uploaded $fileName to $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to upload $fileName with error: " . $exception-
>getMessage();
            exit("Please fix error with file upload before continuing.");
        }

        try {
            $file = $this->s3client->getObject([
                'Bucket' => $this->bucketName,
                'Key' => $fileName,
            ]);
            $body = $file->get('Body');
            $body->rewind();
            echo "Downloaded the file and it begins with: {$body->read(26)}.\n";
        } catch (Exception $exception) {
            echo "Failed to download $fileName from $this->bucketName with error:
 " . $exception->getMessage();
            exit("Please fix error with file downloading before continuing.");
        }

        try {
            $folder = "copied-folder";
            $this->s3client->copyObject([
                'Bucket' => $this->bucketName,
                'CopySource' => "$this->bucketName/$fileName",
                'Key' => "$folder/$fileName-copy",
            ]);
            echo "Copied $fileName to $folder/$fileName-copy.\n";
        } catch (Exception $exception) {
            echo "Failed to copy $fileName with error: " . $exception-
>getMessage();
            exit("Please fix error with object copying before continuing.");
        }

        try {
            $contents = $this->s3client->listObjectsV2([
                'Bucket' => $this->bucketName,
            ]);
            echo "The contents of your bucket are: \n";
            foreach ($contents['Contents'] as $content) {
```

```
                echo $content['Key'] . "\n";
            }
        } catch (Exception $exception) {
            echo "Failed to list objects in $this->bucketName with error: " .
 $exception->getMessage();
            exit("Please fix error with listing objects before continuing.");
        }

        try {
            $objects = [];
            foreach ($contents['Contents'] as $content) {
                $objects[] = [
                    'Key' => $content['Key'],
                ];
            }
            $this->s3client->deleteObjects([
                'Bucket' => $this->bucketName,
                'Delete' => [
                    'Objects' => $objects,
                ],
            ]);
            $check = $this->s3client->listObjectsV2([
                'Bucket' => $this->bucketName,
            ]);
            if (count($check) <= 0) {
                throw new Exception("Bucket wasn't empty.");
            }
            echo "Deleted all objects and folders from $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to delete $fileName from $this->bucketName with error:
 " . $exception->getMessage();
            exit("Please fix error with object deletion before continuing.");
        }

        try {
            $this->s3client->deleteBucket([
                'Bucket' => $this->bucketName,
            ]);
            echo "Deleted bucket $this->bucketName.\n";
        } catch (Exception $exception) {
            echo "Failed to delete $this->bucketName with error: " . $exception-
>getMessage();
            exit("Please fix error with bucket deletion before continuing.");
        }
```

```
        echo "Successfully ran the Amazon S3 with PHP demo.\n";
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for PHP .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import io
import os
import uuid

import boto3
from boto3.s3.transfer import S3UploadFailedError
from botocore.exceptions import ClientError


def do_scenario(s3_resource):
    print("-" * 88)
    print("Welcome to the Amazon S3 getting started demo!")
    print("-" * 88)
```

```
    bucket_name = f"doc-example-bucket-{uuid.uuid4()}"
    bucket = s3_resource.Bucket(bucket_name)
    try:
        bucket.create(
            CreateBucketConfiguration={
                "LocationConstraint": s3_resource.meta.client.meta.region_name
            }
        )
        print(f"Created demo bucket named {bucket.name}.")
    except ClientError as err:
        print(f"Tried and failed to create demo bucket {bucket_name}.")
        print(f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}")
        print(f"\nCan't continue the demo without a bucket!")
        return

    file_name = None
    while file_name is None:
        file_name = input("\nEnter a file you want to upload to your bucket: ")
        if not os.path.exists(file_name):
            print(f"Couldn't find file {file_name}. Are you sure it exists?")
            file_name = None

    obj = bucket.Object(os.path.basename(file_name))
    try:
        obj.upload_file(file_name)
        print(
            f"Uploaded file {file_name} into bucket {bucket.name} with key
 {obj.key}."
        )
    except S3UploadFailedError as err:
        print(f"Couldn't upload file {file_name} to {bucket.name}.")
        print(f"\t{err}")

    answer = input(f"\nDo you want to download {obj.key} into memory (y/n)? ")
    if answer.lower() == "y":
        data = io.BytesIO()
        try:
            obj.download_fileobj(data)
            data.seek(0)
            print(f"Got your object. Here are the first 20 bytes:\n")
            print(f"\t{data.read(20)}")
        except ClientError as err:
```

```
            print(f"Couldn't download {obj.key}.")
            print(
                f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}"
            )

    answer = input(
        f"\nDo you want to copy {obj.key} to a subfolder in your bucket (y/n)? "
    )
    if answer.lower() == "y":
        dest_obj = bucket.Object(f"demo-folder/{obj.key}")
        try:
            dest_obj.copy({"Bucket": bucket.name, "Key": obj.key})
            print(f"Copied {obj.key} to {dest_obj.key}.")
        except ClientError as err:
            print(f"Couldn't copy {obj.key} to {dest_obj.key}.")
            print(
                f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}"
            )

    print("\nYour bucket contains the following objects:")
    try:
        for o in bucket.objects.all():
            print(f"\t{o.key}")
    except ClientError as err:
        print(f"Couldn't list the objects in bucket {bucket.name}.")
        print(f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}")

    answer = input(
        "\nDo you want to delete all of the objects as well as the bucket (y/n)?
 "
    )
    if answer.lower() == "y":
        try:
            bucket.objects.delete()
            bucket.delete()
            print(f"Emptied and deleted bucket {bucket.name}.\n")
        except ClientError as err:
            print(f"Couldn't empty and delete bucket {bucket.name}.")
            print(
                f"\t{err.response['Error']['Code']}:{err.response['Error']
['Message']}"
```

```
            )

    print("Thanks for watching!")
    print("-" * 88)


if __name__ == "__main__":
    do_scenario(boto3.resource("s3"))
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-s3"

# Wraps the getting started scenario actions.
class ScenarioGettingStarted
  attr_reader :s3_resource

  # @param s3_resource [Aws::S3::Resource] An Amazon S3 resource.
  def initialize(s3_resource)
```

```
    @s3_resource = s3_resource
  end

  # Creates a bucket with a random name in the currently configured account and
  # AWS Region.
  #
  # @return [Aws::S3::Bucket] The newly created bucket.
  def create_bucket
    bucket = @s3_resource.create_bucket(
      bucket: "doc-example-bucket-#{Random.uuid}",
      create_bucket_configuration: {
        location_constraint: "us-east-1" # Note: only certain regions permitted
      }
    )
    puts("Created demo bucket named #{bucket.name}.")
  rescue Aws::Errors::ServiceError => e
    puts("Tried and failed to create demo bucket.")
    puts("\t#{e.code}: #{e.message}")
    puts("\nCan't continue the demo without a bucket!")
    raise
  else
    bucket
  end

  # Requests a file name from the user.
  #
  # @return The name of the file.
  def create_file
    File.open("demo.txt", w) { |f| f.write("This is a demo file.") }
  end

  # Uploads a file to an Amazon S3 bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket object representing the upload
destination
  # @return [Aws::S3::Object] The Amazon S3 object that contains the uploaded
file.
  def upload_file(bucket)
    File.open("demo.txt", "w+") { |f| f.write("This is a demo file.") }
    s3_object = bucket.object(File.basename("demo.txt"))
    s3_object.upload_file("demo.txt")
    puts("Uploaded file demo.txt into bucket #{bucket.name} with key
#{s3_object.key}.")
  rescue Aws::Errors::ServiceError => e
```

```
    puts("Couldn't upload file demo.txt to #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  else
    s3_object
  end

  # Downloads an Amazon S3 object to a file.
  #
  # @param s3_object [Aws::S3::Object] The object to download.
  def download_file(s3_object)
    puts("\nDo you want to download #{s3_object.key} to a local file (y/n)? ")
    answer = gets.chomp.downcase
    if answer == "y"
      puts("Enter a name for the downloaded file: ")
      file_name = gets.chomp
      s3_object.download_file(file_name)
      puts("Object #{s3_object.key} successfully downloaded to #{file_name}.")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't download #{s3_object.key}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end

  # Copies an Amazon S3 object to a subfolder within the same bucket.
  #
  # @param source_object [Aws::S3::Object] The source object to copy.
  # @return [Aws::S3::Object, nil] The destination object.
  def copy_object(source_object)
    dest_object = nil
    puts("\nDo you want to copy #{source_object.key} to a subfolder in your
 bucket (y/n)? ")
    answer = gets.chomp.downcase
    if answer == "y"
      dest_object = source_object.bucket.object("demo-folder/
#{source_object.key}")
      dest_object.copy_from(source_object)
      puts("Copied #{source_object.key} to #{dest_object.key}.")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't copy #{source_object.key}.")
    puts("\t#{e.code}: #{e.message}")
    raise
```

```
  else
    dest_object
  end

  # Lists the objects in an Amazon S3 bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket to query.
  def list_objects(bucket)
    puts("\nYour bucket contains the following objects:")
    bucket.objects.each do |obj|
      puts("\t#{obj.key}")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't list the objects in bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end

  # Deletes the objects in an Amazon S3 bucket and deletes the bucket.
  #
  # @param bucket [Aws::S3::Bucket] The bucket to empty and delete.
  def delete_bucket(bucket)
    puts("\nDo you want to delete all of the objects as well as the bucket (y/n)?
 ")
    answer = gets.chomp.downcase
    if answer == "y"
      bucket.objects.batch_delete!
      bucket.delete
      puts("Emptied and deleted bucket #{bucket.name}.\n")
    end
  rescue Aws::Errors::ServiceError => e
    puts("Couldn't empty and delete bucket #{bucket.name}.")
    puts("\t#{e.code}: #{e.message}")
    raise
  end
end

# Runs the Amazon S3 getting started scenario.
def run_scenario(scenario)
  puts("-" * 88)
  puts("Welcome to the Amazon S3 getting started demo!")
  puts("-" * 88)

  bucket = scenario.create_bucket
```

```
  s3_object = scenario.upload_file(bucket)
  scenario.download_file(s3_object)
  scenario.copy_object(s3_object)
  scenario.list_objects(bucket)
  scenario.delete_bucket(bucket)

  puts("Thanks for watching!")
  puts("-" * 88)
rescue Aws::Errors::ServiceError
  puts("Something went wrong with the demo!")
end

run_scenario(ScenarioGettingStarted.new(Aws::S3::Resource.new)) if $PROGRAM_NAME
 == __FILE__
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for Ruby .
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Code pour la caisse binaire qui exécute le scénario.

```
use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::{config::Region, Client};
use s3_service::error::Error;
use uuid::Uuid;


#[tokio::main]
async fn main() -> Result<(), Error> {
    let (region, client, bucket_name, file_name, key, target_key) =
 initialize_variables().await;

    if let Err(e) = run_s3_operations(region, client, bucket_name, file_name,
 key, target_key).await
    {
        println!("{:?}", e);
    };

    Ok(())
}

async fn initialize_variables() -> (Region, Client, String, String, String,
 String) {
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));
    let region = region_provider.region().await.unwrap();

    let shared_config =
 aws_config::from_env().region(region_provider).load().await;
    let client = Client::new(&shared_config);

    let bucket_name = format!("doc-example-bucket-{}", Uuid::new_v4());

    let file_name = "s3/testfile.txt".to_string();
    let key = "test file key name".to_string();
    let target_key = "target_key".to_string();

    (region, client, bucket_name, file_name, key, target_key)
}

async fn run_s3_operations(
    region: Region,
    client: Client,
    bucket_name: String,
```

```
    file_name: String,
    key: String,
    target_key: String,
) -> Result<(), Error> {
    s3_service::create_bucket(&client, &bucket_name, region.as_ref()).await?;
    s3_service::upload_object(&client, &bucket_name, &file_name, &key).await?;
    let _object = s3_service::download_object(&client, &bucket_name, &key).await;
    s3_service::copy_object(&client, &bucket_name, &key, &target_key).await?;
    s3_service::list_objects(&client, &bucket_name).await?;
    s3_service::delete_objects(&client, &bucket_name).await?;
    s3_service::delete_bucket(&client, &bucket_name).await?;

    Ok(())
}
```

Une caisse de bibliothèque avec des actions communes appelées par le code binaire.

```
use aws_sdk_s3::operation::{
    copy_object::{CopyObjectError, CopyObjectOutput},
    create_bucket::{CreateBucketError, CreateBucketOutput},
    get_object::{GetObjectError, GetObjectOutput},
    list_objects_v2::ListObjectsV2Output,
    put_object::{PutObjectError, PutObjectOutput},
};
use aws_sdk_s3::types::{
    BucketLocationConstraint, CreateBucketConfiguration, Delete,
 ObjectIdentifier,
};
use aws_sdk_s3::{error::SdkError, primitives::ByteStream, Client};
use error::Error;
use std::path::Path;
use std::str;

pub mod error;

pub async fn delete_bucket(client: &Client, bucket_name: &str) -> Result<(),
 Error> {
    client.delete_bucket().bucket(bucket_name).send().await?;
    println!("Bucket deleted");
    Ok(())
```

```
}

pub async fn delete_objects(client: &Client, bucket_name: &str) ->
 Result<Vec<String>, Error> {
    let objects = client.list_objects_v2().bucket(bucket_name).send().await?;

    let mut delete_objects: Vec<ObjectIdentifier> = vec![];
    for obj in objects.contents() {
        let obj_id = ObjectIdentifier::builder()
            .set_key(Some(obj.key().unwrap().to_string()))
            .build()
            .map_err(Error::from)?;
        delete_objects.push(obj_id);
    }

    let return_keys = delete_objects.iter().map(|o| o.key.clone()).collect();

    if !delete_objects.is_empty() {
        client
            .delete_objects()
            .bucket(bucket_name)
            .delete(
                Delete::builder()
                    .set_objects(Some(delete_objects))
                    .build()
                    .map_err(Error::from)?,
            )
            .send()
            .await?;
    }

    let objects: ListObjectsV2Output =
 client.list_objects_v2().bucket(bucket_name).send().await?;

    eprintln!("{objects:?}");

    match objects.key_count {
        Some(0) => Ok(return_keys),
        _ => Err(Error::unhandled(
            "There were still objects left in the bucket.",
        )),
    }
}
```

```
pub async fn list_objects(client: &Client, bucket: &str) -> Result<(), Error> {
    let mut response = client
        .list_objects_v2()
        .bucket(bucket.to_owned())
        .max_keys(10) // In this example, go 10 at a time.
        .into_paginator()
        .send();

    while let Some(result) = response.next().await {
        match result {
            Ok(output) => {
                for object in output.contents() {
                    println!(" - {}", object.key().unwrap_or("Unknown"));
                }
            }
            Err(err) => {
                eprintln!("{err:?}")
            }
        }
    }

    Ok(())
}

pub async fn copy_object(
    client: &Client,
    bucket_name: &str,
    object_key: &str,
    target_key: &str,
) -> Result<CopyObjectOutput, SdkError<CopyObjectError>> {
    let mut source_bucket_and_object: String = "".to_owned();
    source_bucket_and_object.push_str(bucket_name);
    source_bucket_and_object.push('/');
    source_bucket_and_object.push_str(object_key);

    client
        .copy_object()
        .copy_source(source_bucket_and_object)
        .bucket(bucket_name)
        .key(target_key)
        .send()
        .await
}
```

```
pub async fn download_object(
    client: &Client,
    bucket_name: &str,
    key: &str,
) -> Result<GetObjectOutput, SdkError<GetObjectError>> {
    client
        .get_object()
        .bucket(bucket_name)
        .key(key)
        .send()
        .await
}

pub async fn upload_object(
    client: &Client,
    bucket_name: &str,
    file_name: &str,
    key: &str,
) -> Result<PutObjectOutput, SdkError<PutObjectError>> {
    let body = ByteStream::from_path(Path::new(file_name)).await;
    client
        .put_object()
        .bucket(bucket_name)
        .key(key)
        .body(body.unwrap())
        .send()
        .await
}

pub async fn create_bucket(
    client: &Client,
    bucket_name: &str,
    region: &str,
) -> Result<CreateBucketOutput, SdkError<CreateBucketError>> {
    let constraint = BucketLocationConstraint::from(region);
    let cfg = CreateBucketConfiguration::builder()
        .location_constraint(constraint)
        .build();
    client
        .create_bucket()
        .create_bucket_configuration(cfg)
        .bucket(bucket_name)
        .send()
        .await
```

```
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Rust API reference.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
DATA(lo_session) = /aws1/cl_rt_session_aws=>create( cv_pfl ).
DATA(lo_s3) = /aws1/cl_s3_factory=>create( lo_session ).

" Create an Amazon Simple Storage Service (Amazon S3) bucket. "
TRY.
    lo_s3->createbucket(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'S3 bucket created.' TYPE 'I'.
  CATCH /aws1/cx_s3_bucketalrdyexists.
    MESSAGE 'Bucket name already exists.' TYPE 'E'.
  CATCH /aws1/cx_s3_bktalrdyownedbyyou.
    MESSAGE 'Bucket already exists and is owned by you.' TYPE 'E'.
ENDTRY.
```

```
"Upload an object to an S3 bucket."
TRY.
    "Get contents of file from application server."
    DATA lv_file_content TYPE xstring.
    OPEN DATASET iv_key FOR INPUT IN BINARY MODE.
    READ DATASET iv_key INTO lv_file_content.
    CLOSE DATASET iv_key.

    lo_s3->putobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
        iv_body = lv_file_content
    ).
    MESSAGE 'Object uploaded to S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.

" Get an object from a bucket. "
TRY.
    DATA(lo_result) = lo_s3->getobject(
                iv_bucket = iv_bucket_name
                iv_key = iv_key
            ).
    DATA(lv_object_data) = lo_result->get_body( ).
    MESSAGE 'Object retrieved from S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" Copy an object to a subfolder in a bucket. "
TRY.
    lo_s3->copyobject(
      iv_bucket = iv_bucket_name
      iv_key = |{ iv_copy_to_folder }/{ iv_key }|
      iv_copysource = |{ iv_bucket_name }/{ iv_key }|
    ).
    MESSAGE 'Object copied to a subfolder.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
  CATCH /aws1/cx_s3_nosuchkey.
```

```
    MESSAGE 'Object key does not exist.' TYPE 'E'.
ENDTRY.

" List objects in the bucket. "
TRY.
    DATA(lo_list) = lo_s3->listobjects(
       iv_bucket = iv_bucket_name
     ).
    MESSAGE 'Retrieved list of objects in S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.
DATA text TYPE string VALUE 'Object List - '.
DATA lv_object_key TYPE /aws1/s3_objectkey.
LOOP AT lo_list->get_contents( ) INTO DATA(lo_object).
  lv_object_key = lo_object->get_key( ).
  CONCATENATE lv_object_key ', ' INTO text.
ENDLOOP.
MESSAGE text TYPE'I'.

" Delete the objects in a bucket. "
TRY.
    lo_s3->deleteobject(
        iv_bucket = iv_bucket_name
        iv_key = iv_key
    ).
    lo_s3->deleteobject(
        iv_bucket = iv_bucket_name
        iv_key = |{ iv_copy_to_folder }/{ iv_key }|
    ).
    MESSAGE 'Objects deleted from S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
ENDTRY.


" Delete the bucket. "
TRY.
    lo_s3->deletebucket(
        iv_bucket = iv_bucket_name
    ).
    MESSAGE 'Deleted S3 bucket.' TYPE 'I'.
  CATCH /aws1/cx_s3_nosuchbucket.
    MESSAGE 'Bucket does not exist.' TYPE 'E'.
```

```
    ENDTRY.
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API du kit AWS SDK pour SAP ABAP.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Swift

Kit SDK pour Swift

> **Note**
>
> Ceci est une documentation préliminaire pour une fonctionnalité en version de prévisualisation. Elle est susceptible d'être modifiée.

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Classe Swift qui gère les appels au SDK pour Swift.

```
import Foundation
import AWSS3
import ClientRuntime
import AWSClientRuntime

/// A class containing all the code that interacts with the AWS SDK for Swift.
```

```
public class ServiceHandler {
    let client: S3Client

    /// Initialize and return a new ``ServiceHandler`` object, which is used to
 drive the AWS calls
    /// used for the example.
    ///
    /// - Returns: A new ``ServiceHandler`` object, ready to be called to
    ///            execute AWS operations.
    public init() async {
        do {
            client = try S3Client(region: "us-east-2")
        } catch {
            print("ERROR: ", dump(error, name: "Initializing S3 client"))
            exit(1)
        }
    }

    /// Create a new user given the specified name.
    ///
    /// - Parameters:
    ///   - name: Name of the bucket to create.
    /// Throws an exception if an error occurs.
    public func createBucket(name: String) async throws {
        let config = S3ClientTypes.CreateBucketConfiguration(
            locationConstraint: .usEast2
        )
        let input = CreateBucketInput(
            bucket: name,
            createBucketConfiguration: config
        )
        _ = try await client.createBucket(input: input)
    }

    /// Delete a bucket.
    /// - Parameter name: Name of the bucket to delete.
    public func deleteBucket(name: String) async throws {
        let input = DeleteBucketInput(
            bucket: name
        )
        _ = try await client.deleteBucket(input: input)
    }

    /// Upload a file from local storage to the bucket.
```

```
    /// - Parameters:
    ///   - bucket: Name of the bucket to upload the file to.
    ///   - key: Name of the file to create.
    ///   - file: Path name of the file to upload.
    public func uploadFile(bucket: String, key: String, file: String) async
throws {
        let fileUrl = URL(fileURLWithPath: file)
        let fileData = try Data(contentsOf: fileUrl)
        let dataStream = ByteStream.from(data: fileData)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }

    /// Create a file in the specified bucket with the given name. The new
    /// file's contents are uploaded from a `Data` object.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket to create a file in.
    ///   - key: Name of the file to create.
    ///   - data: A `Data` object to write into the new file.
    public func createFile(bucket: String, key: String, withData data: Data)
async throws {
        let dataStream = ByteStream.from(data: data)

        let input = PutObjectInput(
            body: dataStream,
            bucket: bucket,
            key: key
        )
        _ = try await client.putObject(input: input)
    }

    /// Download the named file to the given directory on the local device.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket that contains the file to be copied.
    ///   - key: The name of the file to copy from the bucket.
    ///   - to: The path of the directory on the local device where you want to
    ///     download the file.
```

```
    public func downloadFile(bucket: String, key: String, to: String) async
throws {
        let fileUrl = URL(fileURLWithPath: to).appendingPathComponent(key)

        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the data stream object. Return immediately if there isn't one.
        guard let body = output.body,
              let data = try await body.readData() else {
            return
        }
        try data.write(to: fileUrl)
    }

    /// Read the specified file from the given S3 bucket into a Swift
    /// `Data` object.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket containing the file to read.
    ///   - key: Name of the file within the bucket to read.
    ///
    /// - Returns: A `Data` object containing the complete file data.
    public func readFile(bucket: String, key: String) async throws -> Data {
        let input = GetObjectInput(
            bucket: bucket,
            key: key
        )
        let output = try await client.getObject(input: input)

        // Get the stream and return its contents in a `Data` object. If
        // there is no stream, return an empty `Data` object instead.
        guard let body = output.body,
              let data = try await body.readData() else {
            return "".data(using: .utf8)!
        }

        return data
    }

    /// Copy a file from one bucket to another.
```

```
    ///
    /// - Parameters:
    ///   - sourceBucket: Name of the bucket containing the source file.
    ///   - name: Name of the source file.
    ///   - destBucket: Name of the bucket to copy the file into.
    public func copyFile(from sourceBucket: String, name: String, to destBucket:
 String) async throws {
        let srcUrl = ("\(sourceBucket)/
\(name)").addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)

        let input = CopyObjectInput(
            bucket: destBucket,
            copySource: srcUrl,
            key: name
        )
        _ = try await client.copyObject(input: input)
    }

    /// Deletes the specified file from Amazon S3.
    ///
    /// - Parameters:
    ///   - bucket: Name of the bucket containing the file to delete.
    ///   - key: Name of the file to delete.
    ///
    public func deleteFile(bucket: String, key: String) async throws {
        let input = DeleteObjectInput(
            bucket: bucket,
            key: key
        )

        do {
            _ = try await client.deleteObject(input: input)
        } catch {
            throw error
        }
    }

    /// Returns an array of strings, each naming one file in the
    /// specified bucket.
    ///
    /// - Parameter bucket: Name of the bucket to get a file listing for.
    /// - Returns: An array of `String` objects, each giving the name of
    ///            one file contained in the bucket.
    public func listBucketFiles(bucket: String) async throws -> [String] {
```

```
        let input = ListObjectsV2Input(
            bucket: bucket
        )
        let output = try await client.listObjectsV2(input: input)
        var names: [String] = []

        guard let objList = output.contents else {
            return []
        }

        for obj in objList {
            if let objName = obj.key {
                names.append(objName)
            }
        }

        return names
    }
}
```

Un programme de ligne de commande Swift pour gérer les appels du SDK.

```
import Foundation
import ServiceHandler
import ArgumentParser

/// The command-line arguments and options available for this
/// example command.
struct ExampleCommand: ParsableCommand {
    @Argument(help: "Name of the S3 bucket to create")
    var bucketName: String

    @Argument(help: "Pathname of the file to upload to the S3 bucket")
    var uploadSource: String

    @Argument(help: "The name (key) to give the file in the S3 bucket")
    var objName: String

    @Argument(help: "S3 bucket to copy the object to")
    var destBucket: String
```

```
    @Argument(help: "Directory where you want to download the file from the S3
bucket")
    var downloadDir: String

    static var configuration = CommandConfiguration(
        commandName: "s3-basics",
        abstract: "Demonstrates a series of basic AWS S3 functions.",
        discussion: """
        Performs the following Amazon S3 commands:

        * `CreateBucket`
        * `PutObject`
        * `GetObject`
        * `CopyObject`
        * `ListObjects`
        * `DeleteObjects`
        * `DeleteBucket`
        """
    )

    /// Called by ``main()`` to do the actual running of the AWS
    /// example.
    func runAsync() async throws {
        let serviceHandler = await ServiceHandler()

        // 1. Create the bucket.
        print("Creating the bucket \(bucketName)...")
        try await serviceHandler.createBucket(name: bucketName)

        // 2. Upload a file to the bucket.
        print("Uploading the file \(uploadSource)...")
        try await serviceHandler.uploadFile(bucket: bucketName, key: objName,
file: uploadSource)

        // 3. Download the file.
        print("Downloading the file \(objName) to \(downloadDir)...")
        try await serviceHandler.downloadFile(bucket: bucketName, key: objName,
to: downloadDir)

        // 4. Copy the file to another bucket.
        print("Copying the file to the bucket \(destBucket)...")
        try await serviceHandler.copyFile(from: bucketName, name: objName, to:
destBucket)
```

```
        // 5. List the contents of the bucket.

        print("Getting a list of the files in the bucket \(bucketName)")
        let fileList = try await serviceHandler.listBucketFiles(bucket:
 bucketName)
        let numFiles = fileList.count
        if numFiles != 0 {
            print("\(numFiles) file\((numFiles > 1) ? "s" : "") in bucket
 \(bucketName):")
            for name in fileList {
                print("  \(name)")
            }
        } else {
            print("No files found in bucket \(bucketName)")
        }

        // 6. Delete the objects from the bucket.

        print("Deleting the file \(objName) from the bucket \(bucketName)...")
        try await serviceHandler.deleteFile(bucket: bucketName, key: objName)
        print("Deleting the file \(objName) from the bucket \(destBucket)...")
        try await serviceHandler.deleteFile(bucket: destBucket, key: objName)

        // 7. Delete the bucket.
        print("Deleting the bucket \(bucketName)...")
        try await serviceHandler.deleteBucket(name: bucketName)

        print("Done.")
    }
}

//
// Main program entry point.
//
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
```

```
        }
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API du kit SDK AWS pour Swift.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Commencez à chiffrer les objets Amazon S3 à l'aide d'un AWS SDK

L'exemple de code suivant montre comment démarrer avec le chiffrement des objets Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
```

```
    using Amazon.S3.Model;

    /// <summary>
    /// This example shows how to apply client encryption to an object in an
    /// Amazon Simple Storage Service (Amazon S3) bucket.
    /// </summary>
    public class SSEClientEncryption
    {
        public static async Task Main()
        {
            string bucketName = "doc-example-bucket";
            string keyName = "exampleobject.txt";
            string copyTargetKeyName = "examplecopy.txt";

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2.
            IAmazonS3 client = new AmazonS3Client();

            try
            {
                // Create an encryption key.
                Aes aesEncryption = Aes.Create();
                aesEncryption.KeySize = 256;
                aesEncryption.GenerateKey();
                string base64Key = Convert.ToBase64String(aesEncryption.Key);

                // Upload the object.
                PutObjectRequest putObjectRequest = await
UploadObjectAsync(client, bucketName, keyName, base64Key);

                // Download the object and verify that its contents match what
you uploaded.
                await DownloadObjectAsync(client, bucketName, keyName, base64Key,
putObjectRequest);

                // Get object metadata and verify that the object uses AES-256
encryption.
                await GetObjectMetadataAsync(client, bucketName, keyName,
base64Key);

                // Copy both the source and target objects using server-side
encryption with
```

```
                // an encryption key.
                await CopyObjectAsync(client, bucketName, keyName,
copyTargetKeyName, aesEncryption, base64Key);
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }

        /// <summary>
        /// Uploads an object to an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// PutObjectAsync.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket to which
the
        /// object will be uploaded.</param>
        /// <param name="keyName">The name of the object to upload to the Amazon
S3
        /// bucket.</param>
        /// <param name="base64Key">The encryption key.</param>
        /// <returns>The PutObjectRequest object for use by
DownloadObjectAsync.</returns>
        public static async Task<PutObjectRequest> UploadObjectAsync(
            IAmazonS3 client,
            string bucketName,
            string keyName,
            string base64Key)
        {
            PutObjectRequest putObjectRequest = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                ContentBody = "sample text",
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };
            PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
            return putObjectRequest;
        }
```

```
        /// <summary>
        /// Downloads an encrypted object from an Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
 call
        /// GetObjectAsync.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket where the
 object
        /// is located.</param>
        /// <param name="keyName">The name of the Amazon S3 object to download.</
param>
        /// <param name="base64Key">The encryption key used to encrypt the
        /// object.</param>
        /// <param name="putObjectRequest">The PutObjectRequest used to upload
        /// the object.</param>
        public static async Task DownloadObjectAsync(
            IAmazonS3 client,
            string bucketName,
            string keyName,
            string base64Key,
            PutObjectRequest putObjectRequest)
        {
            GetObjectRequest getObjectRequest = new GetObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,

                // Provide encryption information for the object stored in Amazon
 S3.
                ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            using (GetObjectResponse getResponse = await
 client.GetObjectAsync(getObjectRequest))
            using (StreamReader reader = new
 StreamReader(getResponse.ResponseStream))
            {
                string content = reader.ReadToEnd();
                if (string.Compare(putObjectRequest.ContentBody, content) == 0)
                {
                    Console.WriteLine("Object content is same as we uploaded");
```

```
                }
                else
                {
                    Console.WriteLine("Error...Object content is not same.");
                }

                if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
                {
                    Console.WriteLine("Object encryption method is AES256, same
as we set");
                }
                else
                {
                    Console.WriteLine("Error...Object encryption method is not
the same as AES256 we set");
                }
            }
        }

        /// <summary>
        /// Retrieves the metadata associated with an Amazon S3 object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used
        /// to call GetObjectMetadataAsync.</param>
        /// <param name="bucketName">The name of the Amazon S3 bucket containing
the
        /// object for which we want to retrieve metadata.</param>
        /// <param name="keyName">The name of the object for which we wish to
        /// retrieve the metadata.</param>
        /// <param name="base64Key">The encryption key associated with the
        /// object.</param>
        public static async Task GetObjectMetadataAsync(
            IAmazonS3 client,
            string bucketName,
            string keyName,
            string base64Key)
        {
            GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
            {
                BucketName = bucketName,
                Key = keyName,
```

```
                // The object stored in Amazon S3 is encrypted, so provide the
necessary encryption information.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
            Console.WriteLine("The object metadata show encryption method used
is: {0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
        }

        /// <summary>
        /// Copies an encrypted object from one Amazon S3 bucket to another.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// CopyObjectAsync.</param>
        /// <param name="bucketName">The Amazon S3 bucket containing the object
        /// to copy.</param>
        /// <param name="keyName">The name of the object to copy.</param>
        /// <param name="copyTargetKeyName">The Amazon S3 bucket to which the
object
        /// will be copied.</param>
        /// <param name="aesEncryption">The encryption type to use.</param>
        /// <param name="base64Key">The encryption key to use.</param>
        public static async Task CopyObjectAsync(
            IAmazonS3 client,
            string bucketName,
            string keyName,
            string copyTargetKeyName,
            Aes aesEncryption,
            string base64Key)
        {
            aesEncryption.GenerateKey();
            string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

            CopyObjectRequest copyRequest = new CopyObjectRequest
            {
                SourceBucket = bucketName,
                SourceKey = keyName,
                DestinationBucket = bucketName,
                DestinationKey = copyTargetKeyName,
```

```
                // Information about the source object's encryption.
                CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,

                // Information about the target object's encryption.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = copyBase64Key,
            };
            await client.CopyObjectAsync(copyRequest);
        }
    }
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - CopyObject
  - GetObject
  - GetObjectMetadata

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Commencez à utiliser les balises pour les objets Amazon S3 à l'aide d'un AWS SDK

L'exemple de code suivant montre comment démarrer avec des étiquettes pour les objets Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to work with tags in Amazon Simple Storage
/// Service (Amazon S3) objects.
/// </summary>
public class ObjectTag
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket";
        string keyName = "newobject.txt";
        string filePath = @"*** file path ***";

        // Specify your bucket region (an example region is shown).
        RegionEndpoint bucketRegion = RegionEndpoint.USWest2;

        var client = new AmazonS3Client(bucketRegion);
        await PutObjectsWithTagsAsync(client, bucketName, keyName, filePath);
    }

    /// <summary>
    /// This method uploads an object with tags. It then shows the tag
    /// values, changes the tags, and shows the new tags.
    /// </summary>
    /// <param name="client">The Initialized Amazon S3 client object used
    /// to call the methods to create and change an objects tags.</param>
```

```
        /// <param name="bucketName">A string representing the name of the
        /// bucket where the object will be stored.</param>
        /// <param name="keyName">A string representing the key name of the
        /// object to be tagged.</param>
        /// <param name="filePath">The directory location and file name of the
        /// object to be uploaded to the Amazon S3 bucket.</param>
        public static async Task PutObjectsWithTagsAsync(IAmazonS3 client, string
bucketName, string keyName, string filePath)
        {
            try
            {
                // Create an object with tags.
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>
                    {
                        new Tag { Key = "Keyx1", Value = "Value1" },
                        new Tag { Key = "Keyx2", Value = "Value2" },
                    },
                };

                PutObjectResponse response = await
client.PutObjectAsync(putRequest);

                // Now retrieve the new object's tags.
                GetObjectTaggingRequest getTagsRequest = new
GetObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                };

                GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);

                // Display the tag values.
                objectTags.Tagging
                    .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));

                Tagging newTagSet = new Tagging()
```

```
                {
                    TagSet = new List<Tag>
                    {
                        new Tag { Key = "Key3", Value = "Value3" },
                        new Tag { Key = "Key4", Value = "Value4" },
                    },
                };

                PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                    Tagging = newTagSet,
                };

                PutObjectTaggingResponse response2 = await
client.PutObjectTaggingAsync(putObjTagsRequest);

                // Retrieve the tags again and show the values.
                GetObjectTaggingRequest getTagsRequest2 = new
GetObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                };
                GetObjectTaggingResponse objectTags2 = await
client.GetObjectTaggingAsync(getTagsRequest2);

                objectTags2.Tagging
                    .ForEach(t => Console.WriteLine($"Key: {t.Key}, Value:
{t.Value}"));
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine(
                        $"Error: '{ex.Message}'");
            }
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectTaggingà la section Référence des AWS SDK for .NET API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Obtenez la configuration de conservation légale d'un objet Amazon S3 à l'aide d'un AWS SDK

Les exemples de code suivants montrent comment obtenir la configuration de conservation légale d'un compartiment S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/// <summary>
/// Get the legal hold details for an S3 object.
/// </summary>
/// <param name="bucketName">The bucket of the object.</param>
/// <param name="objectKey">The object key.</param>
/// <returns>The object legal hold details.</returns>
public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
    string objectKey)
{
    try
    {
        var request = new GetObjectLegalHoldRequest()
        {
            BucketName = bucketName,
            Key = objectKey
        };
```

```
            var response = await _amazonS3.GetObjectLegalHoldAsync(request);
            Console.WriteLine($"\tObject legal hold for {objectKey} in
{bucketName}: " +
                              $"\n\tStatus: {response.LegalHold.Status}");
            return response.LegalHold;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch legal hold: '{ex.Message}'");
            return new ObjectLockLegalHold();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for .NET API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    // Get the legal hold details for an S3 object.
    public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
        try {
            GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();

            GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
            System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
```

```
                ":\n\tStatus: " + response.legalHold().status());
            return response.legalHold();

        } catch (S3Exception ex) {
            System.out.println("\tUnable to fetch legal hold: '" +
 ex.getMessage() + "'");
        }

        return null;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

import { fileURLToPath } from "url";
import { GetObjectLegalHoldCommand, S3Client } from "@aws-sdk/client-s3";

/**
 * @param {S3Client} client
 * @param {string} bucketName
 * @param {string} objectKey
 */
export const main = async (client, bucketName, objectKey) => {
  const command = new GetObjectLegalHoldCommand({
    Bucket: bucketName,
    Key: objectKey,
    // Optionally, you can provide additional parameters
```

```
    // ExpectedBucketOwner: "ACCOUNT_ID",
    // RequestPayer: "requester",
    // VersionId: "OBJECT_VERSION_ID",
  });

  try {
    const response = await client.send(command);
    console.log(`Legal Hold Status: ${response.LegalHold.Status}`);
  } catch (err) {
    console.error(err);
  }
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main(new S3Client(), "DOC-EXAMPLE-BUCKET", "OBJECT_KEY");
}
```

- Pour plus de détails sur l'API, reportez-vous GetObjectLegalHoldà la section Référence des AWS SDK for JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisez les fonctionnalités de verrouillage d'objets d'Amazon S3 à l'aide d'un AWS SDK

Les exemples de code suivants montrent comment utiliser les fonctionnalités de verrouillage d'objets S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif illustrant les fonctionnalités de verrouillage d'objets d'Amazon S3.

```
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using Microsoft.Extensions.Logging.Console;
using Microsoft.Extensions.Logging.Debug;

namespace S3ObjectLockScenario;

public static class S3ObjectLockWorkflow
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks:
        1. Create test Amazon Simple Storage Service (S3) buckets with different
 lock policies.
        2. Upload sample objects to each bucket.
        3. Set some Legal Hold and Retention Periods on objects and buckets.
        4. Investigate lock policies by viewing settings or attempting to delete
 or overwrite objects.
        5. Clean up objects and buckets.
   */

    public static S3ActionsWrapper _s3ActionsWrapper = null!;
    public static IConfiguration _configuration = null!;
    private static string _resourcePrefix = null!;
    private static string noLockBucketName = null!;
    private static string lockEnabledBucketName = null!;
    private static string retentionAfterCreationBucketName = null!;
    private static List<string> bucketNames = new List<string>();
    private static List<string> fileNames = new List<string>();

    public static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
```

```
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonS3>()
                    .AddTransient<S3ActionsWrapper>()
            )
            .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        ConfigurationSetup();

        ServicesSetup(host);

        try
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Welcome to the Amazon Simple Storage Service (S3)
Object Locking Workflow Scenario.");
            Console.WriteLine(new string('-', 80));
            await Setup(true);

            await DemoActionChoices();

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Cleaning up resources.");
            Console.WriteLine(new string('-', 80));
            await Cleanup(true);

            Console.WriteLine(new string('-', 80));
            Console.WriteLine("Amazon S3 Object Locking Workflow is complete.");
            Console.WriteLine(new string('-', 80));
        }
        catch (Exception ex)
```

```
        {
            Console.WriteLine(new string('-', 80));
            Console.WriteLine($"There was a problem: {ex.Message}");
            await Cleanup(true);
            Console.WriteLine(new string('-', 80));
        }
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _s3ActionsWrapper = host.Services.GetRequiredService<S3ActionsWrapper>();
    }

    /// <summary>
    /// Any setup operations needed.
    /// </summary>
    public static void ConfigurationSetup()
    {
        _resourcePrefix = _configuration["resourcePrefix"] ?? "dotnet-example";

        noLockBucketName = _resourcePrefix + "-no-lock";
        lockEnabledBucketName = _resourcePrefix + "-lock-enabled";
        retentionAfterCreationBucketName = _resourcePrefix + "-retention-after-
creation";

        bucketNames.Add(noLockBucketName);
        bucketNames.Add(lockEnabledBucketName);
        bucketNames.Add(retentionAfterCreationBucketName);
    }

    // <summary>
    /// Deploy necessary resources for the scenario.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Setup(bool interactive)
    {
        Console.WriteLine(
            "\nFor this workflow, we will use the AWS SDK for .NET to create
 several S3\n" +
```

```
            "buckets and files to demonstrate working with S3 locking features.
\n");

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Press Enter when you are ready to start.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("\nS3 buckets can be created either with or without
 object lock enabled.");
        await _s3ActionsWrapper.CreateBucketWithObjectLock(noLockBucketName,
 false);
        await _s3ActionsWrapper.CreateBucketWithObjectLock(lockEnabledBucketName,
 true);
        await
 _s3ActionsWrapper.CreateBucketWithObjectLock(retentionAfterCreationBucketName,
 false);

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("\nA bucket can be configured to use object locking
 with a default retention period.");
        await
 _s3ActionsWrapper.ModifyBucketDefaultRetention(retentionAfterCreationBucketName,
 true,
            ObjectLockRetentionMode.Governance, DateTime.UtcNow.AddDays(1));

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        Console.WriteLine("\nObject lock policies can also be added to existing
 buckets.");
        await _s3ActionsWrapper.EnableObjectLockOnBucket(lockEnabledBucketName);

        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        // Upload some files to the buckets.
        Console.WriteLine("\nNow let's add some test files:");
        var fileName = _configuration["exampleFileName"] ?? "exampleFile.txt";
```

```
        int fileCount = 2;
        // Create the file if it does not already exist.
        if (!File.Exists(fileName))
        {
            await using StreamWriter sw = File.CreateText(fileName);
            await sw.WriteLineAsync(
                "This is a sample file for uploading to a bucket.");
        }

        foreach (var bucketName in bucketNames)
        {
            for (int i = 0; i < fileCount; i++)
            {
                var numberedFileName = Path.GetFileNameWithoutExtension(fileName)
+ i + Path.GetExtension(fileName);
                fileNames.Add(numberedFileName);
                await _s3ActionsWrapper.UploadFileAsync(bucketName,
numberedFileName, fileName);
            }
        }
        Console.WriteLine("Press Enter to continue.");
        if (interactive)
            Console.ReadLine();

        if (!interactive)
            return true;
        Console.WriteLine("\nNow we can set some object lock policies on
individual files:");
        foreach (var bucketName in bucketNames)
        {
            for (int i = 0; i < fileNames.Count; i++)
            {
                // No modifications to the objects in the first bucket.
                if (bucketName != bucketNames[0])
                {
                    var exampleFileName = fileNames[i];
                    switch (i)
                    {
                        case 0:
                            {
                                var question =
                                    $"\nWould you like to add a legal hold to
{exampleFileName} in {bucketName}? (y/n)";
                                if (GetYesNoResponse(question))
```

```
                                {
                                    // Set a legal hold.
                                    await
_s3ActionsWrapper.ModifyObjectLegalHold(bucketName, exampleFileName,
ObjectLockLegalHoldStatus.On);

                                }
                                break;
                            }
                        case 1:
                            {
                                var question =
                                    $"\nWould you like to add a 1 day Governance
retention period to {exampleFileName} in {bucketName}? (y/n)" +
                                    "\nReminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.";
                                if (GetYesNoResponse(question))
                                {
                                    // Set a Governance mode retention period for
1 day.
                                    await
_s3ActionsWrapper.ModifyObjectRetentionPeriod(
                                        bucketName, exampleFileName,
                                        ObjectLockRetentionMode.Governance,
                                        DateTime.UtcNow.AddDays(1));
                                }
                                break;
                            }
                    }
                }
            }
        }
        Console.WriteLine(new string('-', 80));
        return true;
    }

    // <summary>
    /// List all of the current buckets and objects.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>The list of buckets and objects.</returns>
    public static async Task<List<S3ObjectVersion>> ListBucketsAndObjects(bool
interactive)
```

```
    {
        var allObjects = new List<S3ObjectVersion>();
        foreach (var bucketName in bucketNames)
        {
            var objectsInBucket = await
_s3ActionsWrapper.ListBucketObjectsAndVersions(bucketName);
            foreach (var objectKey in objectsInBucket.Versions)
            {
                allObjects.Add(objectKey);
            }
        }

        if (interactive)
        {
            Console.WriteLine("\nCurrent buckets and objects:\n");
            int i = 0;
            foreach (var bucketObject in allObjects)
            {
                i++;
                Console.WriteLine(
                    $"{i}: {bucketObject.Key} \n\tBucket:
{bucketObject.BucketName}\n\tVersion: {bucketObject.VersionId}");
            }
        }

        return allObjects;
    }

    /// <summary>
    /// Present the user with the demo action choices.
    /// </summary>
    /// <returns>Async task.</returns>
    public static async Task<bool> DemoActionChoices()
    {
        var choices = new string[]{
            "List all files in buckets.",
            "Attempt to delete a file.",
            "Attempt to delete a file with retention period bypass.",
            "Attempt to overwrite a file.",
            "View the object and bucket retention settings for a file.",
            "View the legal hold settings for a file.",
            "Finish the workflow."};

        var choice = 0;
```

```
        // Keep asking the user until they choose to move on.
        while (choice != 6)
        {
            Console.WriteLine(new string('-', 80));
            choice = GetChoiceResponse(
                "\nExplore the S3 locking features by selecting one of the
following choices:"
                , choices);
            Console.WriteLine(new string('-', 80));
            switch (choice)
            {
                case 0:
                    {
                        await ListBucketsAndObjects(true);
                        break;
                    }
                case 1:
                    {
                        Console.WriteLine("\nEnter the number of the object to
delete:");
                        var allFiles = await ListBucketsAndObjects(true);
                        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                        await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, false, allFiles[fileChoice].VersionId);
                        break;
                    }
                case 2:
                    {
                        Console.WriteLine("\nEnter the number of the object to
delete:");
                        var allFiles = await ListBucketsAndObjects(true);
                        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                        await
_s3ActionsWrapper.DeleteObjectFromBucket(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, true, allFiles[fileChoice].VersionId);
                        break;
                    }
                case 3:
                    {
                        var allFiles = await ListBucketsAndObjects(true);
```

```
                        Console.WriteLine("\nEnter the number of the object to
overwrite:");
                        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                        // Create the file if it does not already exist.
                        if (!File.Exists(allFiles[fileChoice].Key))
                        {
                            await using StreamWriter sw =
File.CreateText(allFiles[fileChoice].Key);
                            await sw.WriteLineAsync(
                                "This is a sample file for uploading to a
bucket.");
                        }
                        await
_s3ActionsWrapper.UploadFileAsync(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key, allFiles[fileChoice].Key);
                        break;
                    }
                case 4:
                    {
                        var allFiles = await ListBucketsAndObjects(true);
                        Console.WriteLine("\nEnter the number of the object and
bucket to view:");
                        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                        await
_s3ActionsWrapper.GetObjectRetention(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
                        await
_s3ActionsWrapper.GetBucketObjectLockConfiguration(allFiles[fileChoice].BucketName);
                        break;
                    }
                case 5:
                    {
                        var allFiles = await ListBucketsAndObjects(true);
                        Console.WriteLine("\nEnter the number of the object to
view:");
                        var fileChoice = GetChoiceResponse(null,
allFiles.Select(f => f.Key).ToArray());
                        await
_s3ActionsWrapper.GetObjectLegalHold(allFiles[fileChoice].BucketName,
allFiles[fileChoice].Key);
                        break;
                    }
```

```
            }
        }
        return true;
    }

    // <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="interactive">True to run as interactive.</param>
    /// <returns>True if successful.</returns>
    public static async Task<bool> Cleanup(bool interactive)
    {
        Console.WriteLine(new string('-', 80));

        if (!interactive || GetYesNoResponse("Do you want to clean up all files
and buckets? (y/n) "))
        {
            // Remove all locks and delete all buckets and objects.
            var allFiles = await ListBucketsAndObjects(false);
            foreach (var fileInfo in allFiles)
            {
                // Check for a legal hold.
                var legalHold = await
_s3ActionsWrapper.GetObjectLegalHold(fileInfo.BucketName, fileInfo.Key);
                if (legalHold?.Status?.Value == ObjectLockLegalHoldStatus.On)
                {
                    await
_s3ActionsWrapper.ModifyObjectLegalHold(fileInfo.BucketName, fileInfo.Key,
ObjectLockLegalHoldStatus.Off);
                }

                // Check for a retention period.
                var retention = await
_s3ActionsWrapper.GetObjectRetention(fileInfo.BucketName, fileInfo.Key);
                var hasRetentionPeriod = retention?.Mode ==
ObjectLockRetentionMode.Governance && retention.RetainUntilDate >
DateTime.UtcNow.Date;
                await
_s3ActionsWrapper.DeleteObjectFromBucket(fileInfo.BucketName, fileInfo.Key,
hasRetentionPeriod, fileInfo.VersionId);
            }

            foreach (var bucketName in bucketNames)
            {
```

```
                await _s3ActionsWrapper.DeleteBucketByName(bucketName);
            }

        }
        else
        {
            Console.WriteLine(
                "Ok, we'll leave the resources intact.\n" +
                "Don't forget to delete them when you're done with them or you
 might incur unexpected charges."
            );
        }

        Console.WriteLine(new string('-', 80));
        return true;
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null && ynResponse.Equals("y",
 StringComparison.InvariantCultureIgnoreCase);
        return response;
    }

    /// <summary>
    /// Helper method to get a choice response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <param name="choices">The choices to print on the console.</param>
    /// <returns>The index of the selected choice</returns>
    private static int GetChoiceResponse(string? question, string[] choices)
    {
        if (question != null)
        {
            Console.WriteLine(question);
```

```
            for (int i = 0; i < choices.Length; i++)
            {
                Console.WriteLine($"\t{i + 1}. {choices[i]}");
            }
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > choices.Length)
        {
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        return choiceNumber - 1;
    }
}
```

Une classe wrapper pour les fonctions S3.

```
using System.Net;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

namespace S3ObjectLockScenario;

/// <summary>
/// Encapsulate the Amazon S3 operations.
/// </summary>
public class S3ActionsWrapper
{
    private readonly IAmazonS3 _amazonS3;

    /// <summary>
    /// Constructor for the S3ActionsWrapper.
    /// </summary>
    /// <param name="amazonS3">The injected S3 client.</param>
    public S3ActionsWrapper(IAmazonS3 amazonS3, IConfiguration configuration)
    {
        _amazonS3 = amazonS3;
```

```
    }

    /// <summary>
    /// Create a new Amazon S3 bucket with object lock actions.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to create.</param>
    /// <param name="enableObjectLock">True to enable object lock on the
bucket.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> CreateBucketWithObjectLock(string bucketName, bool
enableObjectLock)
    {
        Console.WriteLine($"\tCreating bucket {bucketName} with object lock
{enableObjectLock}.");
        try
        {
            var request = new PutBucketRequest
            {
                BucketName = bucketName,
                UseClientRegion = true,
                ObjectLockEnabledForBucket = enableObjectLock,
            };

            var response = await _amazonS3.PutBucketAsync(request);

            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error creating bucket: '{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Enable object lock on an existing bucket.
    /// </summary>
    /// <param name="bucketName">The name of the bucket to modify.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableObjectLockOnBucket(string bucketName)
    {
        try
        {
            // First, enable Versioning on the bucket.
```

```
            await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig()
                {
                    EnableMfaDelete = false,
                    Status = VersionStatus.Enabled
                }
            });

            var request = new PutObjectLockConfigurationRequest()
            {
                BucketName = bucketName,
                ObjectLockConfiguration = new ObjectLockConfiguration()
                {
                    ObjectLockEnabled = new ObjectLockEnabled("Enabled"),
                },
            };

            var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
            Console.WriteLine($"\tAdded an object lock policy to bucket
{bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"Error modifying object lock: '{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Set or modify a retention period on an object in an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The key of the object.</param>
    /// <param name="retention">The retention mode.</param>
    /// <param name="retainUntilDate">The date retention expires.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyObjectRetentionPeriod(string bucketName,
        string objectKey, ObjectLockRetentionMode retention, DateTime
retainUntilDate)
```

```
    {
        try
        {
            var request = new PutObjectRetentionRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
                Retention = new ObjectLockRetention()
                {
                    Mode = retention,
                    RetainUntilDate = retainUntilDate
                }
            };

            var response = await _amazonS3.PutObjectRetentionAsync(request);
            Console.WriteLine($"\tSet retention for {objectKey} in {bucketName}
until {retainUntilDate:d}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tError modifying retention period:
'{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Set or modify a retention period on an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket to modify.</param>
    /// <param name="retention">The retention mode.</param>
    /// <param name="retainUntilDate">The date for retention until.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyBucketDefaultRetention(string bucketName, bool
enableObjectLock, ObjectLockRetentionMode retention, DateTime retainUntilDate)
    {
        var enabledString = enableObjectLock ? "Enabled" : "Disabled";
        var timeDifference = retainUntilDate.Subtract(DateTime.Now);
        try
        {
            // First, enable Versioning on the bucket.
            await _amazonS3.PutBucketVersioningAsync(new
PutBucketVersioningRequest()
```

```
            {
                BucketName = bucketName,
                VersioningConfig = new S3BucketVersioningConfig()
                {
                    EnableMfaDelete = false,
                    Status = VersionStatus.Enabled
                }
            });

            var request = new PutObjectLockConfigurationRequest()
            {
                BucketName = bucketName,
                ObjectLockConfiguration = new ObjectLockConfiguration()
                {
                    ObjectLockEnabled = new ObjectLockEnabled(enabledString),
                    Rule = new ObjectLockRule()
                    {
                        DefaultRetention = new DefaultRetention()
                        {
                            Mode = retention,
                            Days = timeDifference.Days // Can be specified in
days or years but not both.
                        }
                    }
                }
            };

            var response = await
_amazonS3.PutObjectLockConfigurationAsync(request);
            Console.WriteLine($"\tAdded a default retention to bucket
{bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tError modifying object lock: '{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Get the retention period for an S3 object.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
```

```
    /// <param name="objectKey">The object key.</param>
    /// <returns>The object retention details.</returns>
    public async Task<ObjectLockRetention> GetObjectRetention(string bucketName,
        string objectKey)
    {
        try
        {
            var request = new GetObjectRetentionRequest()
            {
                BucketName = bucketName,
                Key = objectKey
            };

            var response = await _amazonS3.GetObjectRetentionAsync(request);
            Console.WriteLine($"\tObject retention for {objectKey} in
{bucketName}: " +
                              $"\n\t{response.Retention.Mode} until
{response.Retention.RetainUntilDate:d}.");
            return response.Retention;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch object lock retention:
'{ex.Message}'");
            return new ObjectLockRetention();
        }
    }

    /// <summary>
    /// Set or modify a legal hold on an object in an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The key of the object.</param>
    /// <param name="holdStatus">The On or Off status for the legal hold.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> ModifyObjectLegalHold(string bucketName,
        string objectKey, ObjectLockLegalHoldStatus holdStatus)
    {
        try
        {
            var request = new PutObjectLegalHoldRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
```

```
                LegalHold = new ObjectLockLegalHold()
                {
                    Status = holdStatus
                }
            };

            var response = await _amazonS3.PutObjectLegalHoldAsync(request);
            Console.WriteLine($"\tModified legal hold for {objectKey} in
{bucketName}.");
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tError modifying legal hold: '{ex.Message}'");
            return false;
        }
    }

    /// <summary>
    /// Get the legal hold details for an S3 object.
    /// </summary>
    /// <param name="bucketName">The bucket of the object.</param>
    /// <param name="objectKey">The object key.</param>
    /// <returns>The object legal hold details.</returns>
    public async Task<ObjectLockLegalHold> GetObjectLegalHold(string bucketName,
        string objectKey)
    {
        try
        {
            var request = new GetObjectLegalHoldRequest()
            {
                BucketName = bucketName,
                Key = objectKey
            };

            var response = await _amazonS3.GetObjectLegalHoldAsync(request);
            Console.WriteLine($"\tObject legal hold for {objectKey} in
{bucketName}: " +
                              $"\n\tStatus: {response.LegalHold.Status}");
            return response.LegalHold;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch legal hold: '{ex.Message}'");
```

```
            return new ObjectLockLegalHold();
        }
    }

    /// <summary>
    /// Get the object lock configuration details for an S3 bucket.
    /// </summary>
    /// <param name="bucketName">The bucket to get details.</param>
    /// <returns>The bucket's object lock configuration details.</returns>
    public async Task<ObjectLockConfiguration>
GetBucketObjectLockConfiguration(string bucketName)
    {
        try
        {
            var request = new GetObjectLockConfigurationRequest()
            {
                BucketName = bucketName
            };

            var response = await
_amazonS3.GetObjectLockConfigurationAsync(request);
            Console.WriteLine($"\tBucket object lock config for {bucketName} in
{bucketName}: " +
                              $"\n\tEnabled:
{response.ObjectLockConfiguration.ObjectLockEnabled}" +
                              $"\n\tRule:
{response.ObjectLockConfiguration.Rule?.DefaultRetention}");

            return response.ObjectLockConfiguration;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to fetch object lock config:
'{ex.Message}'");
            return new ObjectLockConfiguration();
        }
    }

    /// <summary>
    /// Upload a file from the local computer to an Amazon S3 bucket.
    /// </summary>
    /// <param name="bucketName">The Amazon S3 bucket to use.</param>
    /// <param name="objectName">The object to upload.</param>
```

```
   /// <param name="filePath">The path, including file name, of the object to
upload.</param>
   /// <returns>True if success.<returns>
   public async Task<bool> UploadFileAsync(string bucketName, string objectName,
string filePath)
   {
       var request = new PutObjectRequest
       {
           BucketName = bucketName,
           Key = objectName,
           FilePath = filePath,
           ChecksumAlgorithm = ChecksumAlgorithm.SHA256
       };

       var response = await _amazonS3.PutObjectAsync(request);
       if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
       {
           Console.WriteLine($"\tSuccessfully uploaded {objectName} to
{bucketName}.");
           return true;
       }
       else
       {
           Console.WriteLine($"\tCould not upload {objectName} to
{bucketName}.");
           return false;
       }
   }

   /// <summary>
   /// List bucket objects and versions.
   /// </summary>
   /// <param name="bucketName">The Amazon S3 bucket to use.</param>
   /// <returns>The list of objects and versions.</returns>
   public async Task<ListVersionsResponse> ListBucketObjectsAndVersions(string
bucketName)
   {
       var request = new ListVersionsRequest()
       {
           BucketName = bucketName
       };

       var response = await _amazonS3.ListVersionsAsync(request);
       return response;
```

```
    }

    /// <summary>
    /// Delete an object from a specific bucket.
    /// </summary>
    /// <param name="bucketName">The Amazon S3 bucket to use.</param>
    /// <param name="objectKey">The key of the object to delete.</param>
    /// <param name="hasRetention">True if the object has retention settings.</
param>
    /// <param name="versionId">Optional versionId.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteObjectFromBucket(string bucketName, string
 objectKey, bool hasRetention, string? versionId = null)
    {
        try
        {
            var request = new DeleteObjectRequest()
            {
                BucketName = bucketName,
                Key = objectKey,
                VersionId = versionId,
            };
            if (hasRetention)
            {
                // Set the BypassGovernanceRetention header
                // if the file has retention settings.
                request.BypassGovernanceRetention = true;
            }
            await _amazonS3.DeleteObjectAsync(request);
            Console.WriteLine(
                $"Deleted {objectKey} in {bucketName}.");
            return true;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to delete object {objectKey} in bucket
 {bucketName}: " + ex.Message);
            return false;
        }
    }

    /// <summary>
    /// Delete a specific bucket.
    /// </summary>
```

```
    /// <param name="bucketName">The Amazon S3 bucket to use.</param>
    /// <param name="objectKey">The key of the object to delete.</param>
    /// <param name="versionId">Optional versionId.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteBucketByName(string bucketName)
    {
        try
        {
            var request = new DeleteBucketRequest() { BucketName = bucketName, };
            var response = await _amazonS3.DeleteBucketAsync(request);
            Console.WriteLine($"\tDelete for {bucketName} complete.");
            return response.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine($"\tUnable to delete bucket {bucketName}: " +
 ex.Message);
            return false;
        }

    }

}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif illustrant les fonctionnalités de verrouillage d'objets d'Amazon S3.

```
// ObjectLockScenario contains the steps to run the S3 Object Lock workflow.
type ObjectLockScenario struct {
 questioner demotools.IQuestioner
 resources  Resources
 s3Actions  *actions.S3Actions
 sdkConfig  aws.Config
}

// NewObjectLockScenario constructs a new ObjectLockScenario instance.
func NewObjectLockScenario(sdkConfig aws.Config, questioner
 demotools.IQuestioner) ObjectLockScenario {
 scenario := ObjectLockScenario{
  questioner: questioner,
  resources:  Resources{},
  s3Actions:  &actions.S3Actions{S3Client: s3.NewFromConfig(sdkConfig)},
  sdkConfig:  sdkConfig,
 }
 scenario.s3Actions.S3Manager = manager.NewUploader(scenario.s3Actions.S3Client)
 scenario.resources.init(scenario.s3Actions, questioner)
 return scenario
}

type nameLocked struct {
 name   string
 locked bool
}

var createInfo = []nameLocked{
```

```
 {"standard-bucket", false},
 {"lock-bucket", true},
 {"retention-bucket", false},
}

// CreateBuckets creates the S3 buckets required for the workflow.
func (scenario *ObjectLockScenario) CreateBuckets(ctx context.Context) {
 log.Println("Let's create some S3 buckets to use for this workflow.")
 success := false
 for !success {
  prefix := scenario.questioner.Ask(
   "This example creates three buckets. Enter a prefix to name your buckets
 (remember bucket names must be globally unique):")

  for _, info := range createInfo {
   bucketName, err := scenario.s3Actions.CreateBucketWithLock(ctx,
 fmt.Sprintf("%s.%s", prefix, info.name), scenario.sdkConfig.Region, info.locked)
   if err != nil {
    switch err.(type) {
    case *types.BucketAlreadyExists, *types.BucketAlreadyOwnedByYou:
     log.Printf("Couldn't create bucket %s.\n", bucketName)
    default:
     panic(err)
    }
    break
   }
   scenario.resources.demoBuckets[info.name] = &DemoBucket{
    name:       bucketName,
    objectKeys: []string{},
   }
   log.Printf("Created bucket %s.\n", bucketName)
  }

  if len(scenario.resources.demoBuckets) < len(createInfo) {
   scenario.resources.deleteBuckets(ctx)
  } else {
   success = true
  }
 }

 log.Println("S3 buckets created.")
 log.Println(strings.Repeat("-", 88))
}
```

```
// EnableLockOnBucket enables object locking on an existing bucket.
func (scenario *ObjectLockScenario) EnableLockOnBucket(ctx context.Context) {
 log.Println("\nA bucket can be configured to use object locking.")
 scenario.questioner.Ask("Press Enter to continue.")

 var err error
 bucket := scenario.resources.demoBuckets["retention-bucket"]
 err = scenario.s3Actions.EnableObjectLockOnBucket(ctx, bucket.name)
 if err != nil {
  switch err.(type) {
  case *types.NoSuchBucket:
   log.Printf("Couldn't enable object locking on bucket %s.\n", bucket.name)
  default:
   panic(err)
  }
 } else {
  log.Printf("Object locking enabled on bucket %s.", bucket.name)
 }

 log.Println(strings.Repeat("-", 88))
}

// SetDefaultRetentionPolicy sets a default retention governance policy on a
 bucket.
func (scenario *ObjectLockScenario) SetDefaultRetentionPolicy(ctx
 context.Context) {
 log.Println("\nA bucket can be configured to use object locking with a default
 retention period.")

 bucket := scenario.resources.demoBuckets["retention-bucket"]
 retentionPeriod := scenario.questioner.AskInt("Enter the default retention
 period in days: ")
 err := scenario.s3Actions.ModifyDefaultBucketRetention(ctx,
 bucket.name, types.ObjectLockEnabledEnabled, int32(retentionPeriod),
 types.ObjectLockRetentionModeGovernance)
 if err != nil {
  switch err.(type) {
  case *types.NoSuchBucket:
   log.Printf("Couldn't configure a default retention period on bucket %s.\n",
 bucket.name)
  default:
   panic(err)
  }
 } else {
```

```
  log.Printf("Default retention policy set on bucket %s with %d day retention
 period.", bucket.name, retentionPeriod)
  bucket.retentionEnabled = true
 }

 log.Println(strings.Repeat("-", 88))
}

// UploadTestObjects uploads test objects to the S3 buckets.
func (scenario *ObjectLockScenario) UploadTestObjects(ctx context.Context) {
 log.Println("Uploading test objects to S3 buckets.")

 for _, info := range createInfo {
  bucket := scenario.resources.demoBuckets[info.name]
  for i := 0; i < 2; i++ {
   key, err := scenario.s3Actions.UploadObject(ctx, bucket.name,
 fmt.Sprintf("example-%d", i),
    fmt.Sprintf("Example object content #%d in bucket %s.", i, bucket.name))
   if err != nil {
    switch err.(type) {
    case *types.NoSuchBucket:
     log.Printf("Couldn't upload %s to bucket %s.\n", key, bucket.name)
    default:
     panic(err)
    }
   } else {
    log.Printf("Uploaded %s to bucket %s.\n", key, bucket.name)
    bucket.objectKeys = append(bucket.objectKeys, key)
   }
  }
 }

 scenario.questioner.Ask("Test objects uploaded. Press Enter to continue.")
 log.Println(strings.Repeat("-", 88))
}

// SetObjectLockConfigurations sets object lock configurations on the test
 objects.
func (scenario *ObjectLockScenario) SetObjectLockConfigurations(ctx
 context.Context) {
 log.Println("Now let's set object lock configurations on individual objects.")

 buckets := []*DemoBucket{scenario.resources.demoBuckets["lock-bucket"],
 scenario.resources.demoBuckets["retention-bucket"]}
```

```
for _, bucket := range buckets {
 for index, objKey := range bucket.objectKeys {
  switch index {
  case 0:
   if scenario.questioner.AskBool(fmt.Sprintf("\nDo you want to add a legal hold
to %s in %s (y/n)? ", objKey, bucket.name), "y") {
    err := scenario.s3Actions.PutObjectLegalHold(ctx, bucket.name, objKey, "",
types.ObjectLockLegalHoldStatusOn)
    if err != nil {
     switch err.(type) {
     case *types.NoSuchKey:
      log.Printf("Couldn't set legal hold on %s.\n", objKey)
     default:
      panic(err)
     }
    } else {
     log.Printf("Legal hold set on %s.\n", objKey)
    }
   }
  case 1:
   q := fmt.Sprintf("\nDo you want to add a 1 day Governance retention period to
%s in %s?\n"+
    "Reminder: Only a user with the s3:BypassGovernanceRetention permission is
able to delete this object\n"+
    "or its bucket until the retention period has expired. (y/n) ", objKey,
bucket.name)
   if scenario.questioner.AskBool(q, "y") {
    err := scenario.s3Actions.PutObjectRetention(ctx, bucket.name, objKey,
types.ObjectLockRetentionModeGovernance, 1)
    if err != nil {
     switch err.(type) {
     case *types.NoSuchKey:
      log.Printf("Couldn't set retention period on %s in %s.\n", objKey,
bucket.name)
     default:
      panic(err)
     }
    } else {
     log.Printf("Retention period set to 1 for %s.", objKey)
     bucket.retentionEnabled = true
    }
   }
  }
 }
```

```
 }
 log.Println(strings.Repeat("-", 88))
}

const (
 ListAll = iota
 DeleteObject
 DeleteRetentionObject
 OverwriteObject
 ViewRetention
 ViewLegalHold
 Finish
)

// InteractWithObjects allows the user to interact with the objects and test the
 object lock configurations.
func (scenario *ObjectLockScenario) InteractWithObjects(ctx context.Context) {
 log.Println("Now you can interact with the objects to explore the object lock
 configurations.")
 interactiveChoices := []string{
  "List all objects and buckets.",
  "Attempt to delete an object.",
  "Attempt to delete an object with retention period bypass.",
  "Attempt to overwrite a file.",
  "View the retention settings for an object.",
  "View the legal hold settings for an object.",
  "Finish the workflow."}

 choice := ListAll
 for choice != Finish {
  objList := scenario.GetAllObjects(ctx)
  objChoices := scenario.makeObjectChoiceList(objList)
  choice = scenario.questioner.AskChoice("Choose an action from the menu:\n",
 interactiveChoices)
  switch choice {
  case ListAll:
   log.Println("The current objects in the example buckets are:")
   for _, objChoice := range objChoices {
    log.Println("\t", objChoice)
   }
  case DeleteObject, DeleteRetentionObject:
   objChoice := scenario.questioner.AskChoice("Enter the number of the object to
 delete:\n", objChoices)
   obj := objList[objChoice]
```

```
  deleted, err := scenario.s3Actions.DeleteObject(ctx, obj.bucket, obj.key,
obj.versionId, choice == DeleteRetentionObject)
  if err != nil {
   switch err.(type) {
   case *types.NoSuchKey:
    log.Println("Nothing to delete.")
   default:
    panic(err)
   }
  } else if deleted {
   log.Printf("Object %s deleted.\n", obj.key)
  }
 case OverwriteObject:
  objChoice := scenario.questioner.AskChoice("Enter the number of the object to
overwrite:\n", objChoices)
  obj := objList[objChoice]
  _, err := scenario.s3Actions.UploadObject(ctx, obj.bucket, obj.key,
fmt.Sprintf("New content in object %s.", obj.key))
  if err != nil {
   switch err.(type) {
   case *types.NoSuchBucket:
    log.Println("Couldn't upload to nonexistent bucket.")
   default:
    panic(err)
   }
  } else {
   log.Printf("Uploaded new content to object %s.\n", obj.key)
  }
 case ViewRetention:
  objChoice := scenario.questioner.AskChoice("Enter the number of the object to
view:\n", objChoices)
  obj := objList[objChoice]
  retention, err := scenario.s3Actions.GetObjectRetention(ctx, obj.bucket,
obj.key)
  if err != nil {
   switch err.(type) {
   case *types.NoSuchKey:
    log.Printf("Can't get retention configuration for %s.\n", obj.key)
   default:
    panic(err)
   }
  } else if retention != nil {
   log.Printf("Object %s has retention mode %s until %v.\n", obj.key,
retention.Mode, retention.RetainUntilDate)
```

```
   } else {
    log.Printf("Object %s does not have object retention configured.\n", obj.key)
   }
  case ViewLegalHold:
   objChoice := scenario.questioner.AskChoice("Enter the number of the object to
 view:\n", objChoices)
   obj := objList[objChoice]
   legalHold, err := scenario.s3Actions.GetObjectLegalHold(ctx, obj.bucket,
 obj.key, obj.versionId)
   if err != nil {
    switch err.(type) {
    case *types.NoSuchKey:
     log.Printf("Can't get legal hold configuration for %s.\n", obj.key)
    default:
     panic(err)
    }
   } else if legalHold != nil {
    log.Printf("Object %s has legal hold %v.", obj.key, *legalHold)
   } else {
    log.Printf("Object %s does not have legal hold configured.", obj.key)
   }
  case Finish:
   log.Println("Let's clean up.")
  }
  log.Println(strings.Repeat("-", 88))
 }
}

type BucketKeyVersionId struct {
 bucket    string
 key       string
 versionId string
}

// GetAllObjects gets the object versions in the example S3 buckets and returns
 them in a flattened list.
func (scenario *ObjectLockScenario) GetAllObjects(ctx context.Context)
 []BucketKeyVersionId {
 var objectList []BucketKeyVersionId
 for _, info := range createInfo {
  bucket := scenario.resources.demoBuckets[info.name]
  versions, err := scenario.s3Actions.ListObjectVersions(ctx, bucket.name)
  if err != nil {
   switch err.(type) {
```

```
   case *types.NoSuchBucket:
    log.Printf("Couldn't get object versions for %s.\n", bucket.name)
   default:
    panic(err)
   }
  } else {
   for _, version := range versions {
    objectList = append(objectList,
     BucketKeyVersionId{bucket: bucket.name, key: *version.Key, versionId:
 *version.VersionId})
   }
  }
 }
 return objectList
}

// makeObjectChoiceList makes the object version list into a list of strings that
 are displayed
// as choices.
func (scenario *ObjectLockScenario) makeObjectChoiceList(bucketObjects
 []BucketKeyVersionId) []string {
 choices := make([]string, len(bucketObjects))
 for i := 0; i < len(bucketObjects); i++ {
  choices[i] = fmt.Sprintf("%s in %s with VersionId %s.",
   bucketObjects[i].key, bucketObjects[i].bucket, bucketObjects[i].versionId)
 }
 return choices
}

// Run runs the S3 Object Lock workflow scenario.
func (scenario *ObjectLockScenario) Run(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
   _, isMock := scenario.questioner.(*demotools.MockQuestioner)
   if isMock || scenario.questioner.AskBool("Do you want to see the full error
 message (y/n)?", "y") {
    log.Println(r)
   }
   scenario.resources.Cleanup(ctx)
  }
 }()

 log.Println(strings.Repeat("-", 88))
```

```
 log.Println("Welcome to the Amazon S3 Object Lock Workflow Scenario.")
 log.Println(strings.Repeat("-", 88))

 scenario.CreateBuckets(ctx)
 scenario.EnableLockOnBucket(ctx)
 scenario.SetDefaultRetentionPolicy(ctx)
 scenario.UploadTestObjects(ctx)
 scenario.SetObjectLockConfigurations(ctx)
 scenario.InteractWithObjects(ctx)

 scenario.resources.Cleanup(ctx)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}
```

Définissez une structure qui enveloppe les actions S3 utilisées dans cet exemple.

```
// S3Actions wraps S3 service actions.
type S3Actions struct {
 S3Client  *s3.Client
 S3Manager *manager.Uploader
}



// CreateBucketWithLock creates a new S3 bucket with optional object locking
 enabled
// and waits for the bucket to exist before returning.
func (actor S3Actions) CreateBucketWithLock(ctx context.Context, bucket string,
 region string, enableObjectLock bool) (string, error) {
 input := &s3.CreateBucketInput{
  Bucket: aws.String(bucket),
  CreateBucketConfiguration: &types.CreateBucketConfiguration{
   LocationConstraint: types.BucketLocationConstraint(region),
  },
 }

 if enableObjectLock {
```

```
  input.ObjectLockEnabledForBucket = aws.Bool(true)
 }

 _, err := actor.S3Client.CreateBucket(ctx, input)
 if err != nil {
  var owned *types.BucketAlreadyOwnedByYou
  var exists *types.BucketAlreadyExists
  if errors.As(err, &owned) {
   log.Printf("You already own bucket %s.\n", bucket)
   err = owned
  } else if errors.As(err, &exists) {
   log.Printf("Bucket %s already exists.\n", bucket)
   err = exists
  }
 } else {
  err = s3.NewBucketExistsWaiter(actor.S3Client).Wait(
   ctx, &s3.HeadBucketInput{Bucket: aws.String(bucket)}, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for bucket %s to exist.\n", bucket)
  }
 }

 return bucket, err
}



// GetObjectLegalHold retrieves the legal hold status for an S3 object.
func (actor S3Actions) GetObjectLegalHold(ctx context.Context, bucket string, key
 string, versionId string) (*types.ObjectLockLegalHoldStatus, error) {
 var status *types.ObjectLockLegalHoldStatus
 input := &s3.GetObjectLegalHoldInput{
  Bucket:    aws.String(bucket),
  Key:       aws.String(key),
  VersionId: aws.String(versionId),
 }

 output, err := actor.S3Client.GetObjectLegalHold(ctx, input)
 if err != nil {
  var noSuchKeyErr *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noSuchKeyErr) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noSuchKeyErr
```

```
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
   case "NoSuchObjectLockConfiguration":
    log.Printf("Object %s does not have an object lock configuration.\n", key)
    err = nil
   case "InvalidRequest":
    log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
    err = nil
   }
  }
 } else {
  status = &output.LegalHold.Status
 }

 return status, err
}



// GetObjectLockConfiguration retrieves the object lock configuration for an S3
 bucket.
func (actor S3Actions) GetObjectLockConfiguration(ctx context.Context, bucket
 string) (*types.ObjectLockConfiguration, error) {
 var lockConfig *types.ObjectLockConfiguration
 input := &s3.GetObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
 }

 output, err := actor.S3Client.GetObjectLockConfiguration(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  } else if errors.As(err, &apiErr) && apiErr.ErrorCode() ==
 "ObjectLockConfigurationNotFoundError" {
   log.Printf("Bucket %s does not have an object lock configuration.\n", bucket)
   err = nil
  }
 } else {
  lockConfig = output.ObjectLockConfiguration
 }
```

```
 return lockConfig, err
}



// GetObjectRetention retrieves the object retention configuration for an S3
 object.
func (actor S3Actions) GetObjectRetention(ctx context.Context, bucket string, key
 string) (*types.ObjectLockRetention, error) {
 var retention *types.ObjectLockRetention
 input := &s3.GetObjectRetentionInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
 }

 output, err := actor.S3Client.GetObjectRetention(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
   case "NoSuchObjectLockConfiguration":
    err = nil
   case "InvalidRequest":
    log.Printf("Bucket %s does not have locking enabled.", bucket)
    err = nil
   }
  }
 } else {
  retention = output.Retention
 }

 return retention, err
}



// PutObjectLegalHold sets the legal hold configuration for an S3 object.
func (actor S3Actions) PutObjectLegalHold(ctx context.Context, bucket string, key
 string, versionId string, legalHoldStatus types.ObjectLockLegalHoldStatus) error
 {
```

```
 input := &s3.PutObjectLegalHoldInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
  LegalHold: &types.ObjectLockLegalHold{
   Status: legalHoldStatus,
  },
 }
 if versionId != "" {
  input.VersionId = aws.String(versionId)
 }

 _, err := actor.S3Client.PutObjectLegalHold(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  }
 }

 return err
}



// ModifyDefaultBucketRetention modifies the default retention period of an
 existing bucket.
func (actor S3Actions) ModifyDefaultBucketRetention(
 ctx context.Context, bucket string, lockMode types.ObjectLockEnabled,
 retentionPeriod int32, retentionMode types.ObjectLockRetentionMode) error {

 input := &s3.PutObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
  ObjectLockConfiguration: &types.ObjectLockConfiguration{
   ObjectLockEnabled: lockMode,
   Rule: &types.ObjectLockRule{
    DefaultRetention: &types.DefaultRetention{
     Days: aws.Int32(retentionPeriod),
     Mode: retentionMode,
    },
   },
  },
 }
 _, err := actor.S3Client.PutObjectLockConfiguration(ctx, input)
```

```
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 }

 return err
}



// EnableObjectLockOnBucket enables object locking on an existing bucket.
func (actor S3Actions) EnableObjectLockOnBucket(ctx context.Context, bucket
 string) error {
 // Versioning must be enabled on the bucket before object locking is enabled.
 verInput := &s3.PutBucketVersioningInput{
  Bucket: aws.String(bucket),
  VersioningConfiguration: &types.VersioningConfiguration{
   MFADelete: types.MFADeleteDisabled,
   Status:    types.BucketVersioningStatusEnabled,
  },
 }
 _, err := actor.S3Client.PutBucketVersioning(ctx, verInput)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
  return err
 }

 input := &s3.PutObjectLockConfigurationInput{
  Bucket: aws.String(bucket),
  ObjectLockConfiguration: &types.ObjectLockConfiguration{
   ObjectLockEnabled: types.ObjectLockEnabledEnabled,
  },
 }
 _, err = actor.S3Client.PutObjectLockConfiguration(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
```

```
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 }

 return err
}



// PutObjectRetention sets the object retention configuration for an S3 object.
func (actor S3Actions) PutObjectRetention(ctx context.Context, bucket string, key
 string, retentionMode types.ObjectLockRetentionMode, retentionPeriodDays int32)
 error {
 input := &s3.PutObjectRetentionInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
  Retention: &types.ObjectLockRetention{
   Mode:            retentionMode,
   RetainUntilDate: aws.Time(time.Now().AddDate(0, 0, int(retentionPeriodDays))),
  },
  BypassGovernanceRetention: aws.Bool(true),
 }

 _, err := actor.S3Client.PutObjectRetention(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in bucket %s.\n", key, bucket)
   err = noKey
  }
 }

 return err
}



// UploadObject uses the S3 upload manager to upload an object to a bucket.
func (actor S3Actions) UploadObject(ctx context.Context, bucket string, key
 string, contents string) (string, error) {
 var outKey string
 input := &s3.PutObjectInput{
  Bucket:            aws.String(bucket),
```

```
  Key:               aws.String(key),
  Body:              bytes.NewReader([]byte(contents)),
  ChecksumAlgorithm: types.ChecksumAlgorithmSha256,
 }
 output, err := actor.S3Manager.Upload(ctx, input)
 if err != nil {
  var noBucket *types.NoSuchBucket
  if errors.As(err, &noBucket) {
   log.Printf("Bucket %s does not exist.\n", bucket)
   err = noBucket
  }
 } else {
  err := s3.NewObjectExistsWaiter(actor.S3Client).Wait(ctx, &s3.HeadObjectInput{
   Bucket: aws.String(bucket),
   Key:    aws.String(key),
  }, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for object %s to exist in %s.\n", key,
 bucket)
  } else {
   outKey = *output.Key
  }
 }
 return outKey, err
}



// ListObjectVersions lists all versions of all objects in a bucket.
func (actor S3Actions) ListObjectVersions(ctx context.Context, bucket string)
 ([]types.ObjectVersion, error) {
 var err error
 var output *s3.ListObjectVersionsOutput
 var versions []types.ObjectVersion
 input := &s3.ListObjectVersionsInput{Bucket: aws.String(bucket)}
 versionPaginator := s3.NewListObjectVersionsPaginator(actor.S3Client, input)
 for versionPaginator.HasMorePages() {
  output, err = versionPaginator.NextPage(ctx)
  if err != nil {
   var noBucket *types.NoSuchBucket
   if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
   }
```

```
   break
  } else {
   versions = append(versions, output.Versions...)
  }
 }
 return versions, err
}



// DeleteObject deletes an object from a bucket.
func (actor S3Actions) DeleteObject(ctx context.Context, bucket string, key
 string, versionId string, bypassGovernance bool) (bool, error) {
 deleted := false
 input := &s3.DeleteObjectInput{
  Bucket: aws.String(bucket),
  Key:    aws.String(key),
 }
 if versionId != "" {
  input.VersionId = aws.String(versionId)
 }
 if bypassGovernance {
  input.BypassGovernanceRetention = aws.Bool(true)
 }
 _, err := actor.S3Client.DeleteObject(ctx, input)
 if err != nil {
  var noKey *types.NoSuchKey
  var apiErr *smithy.GenericAPIError
  if errors.As(err, &noKey) {
   log.Printf("Object %s does not exist in %s.\n", key, bucket)
   err = noKey
  } else if errors.As(err, &apiErr) {
   switch apiErr.ErrorCode() {
   case "AccessDenied":
    log.Printf("Access denied: cannot delete object %s from %s.\n", key, bucket)
    err = nil
   case "InvalidArgument":
    if bypassGovernance {
     log.Printf("You cannot specify bypass governance on a bucket without lock
 enabled.")
     err = nil
    }
   }
  }
```

```
 } else {
  deleted = true
 }
 return deleted, err
}



// DeleteObjects deletes a list of objects from a bucket.
func (actor S3Actions) DeleteObjects(ctx context.Context, bucket string, objects
 []types.ObjectIdentifier, bypassGovernance bool) error {
 if len(objects) == 0 {
  return nil
 }

 input := s3.DeleteObjectsInput{
  Bucket: aws.String(bucket),
  Delete: &types.Delete{
   Objects: objects,
   Quiet:   aws.Bool(true),
  },
 }
 if bypassGovernance {
  input.BypassGovernanceRetention = aws.Bool(true)
 }
 delOut, err := actor.S3Client.DeleteObjects(ctx, &input)
 if err != nil || len(delOut.Errors) > 0 {
  log.Printf("Error deleting objects from bucket %s.\n", bucket)
  if err != nil {
   var noBucket *types.NoSuchBucket
   if errors.As(err, &noBucket) {
    log.Printf("Bucket %s does not exist.\n", bucket)
    err = noBucket
   }
  } else if len(delOut.Errors) > 0 {
   for _, outErr := range delOut.Errors {
    log.Printf("%s: %s\n", *outErr.Key, *outErr.Message)
   }
   err = fmt.Errorf("%s", *delOut.Errors[0].Message)
  }
 }
 return err
}
```

Nettoyez les ressources.

```
// DemoBucket contains metadata for buckets used in this example.
type DemoBucket struct {
 name             string
 legalHold        bool
 retentionEnabled bool
 objectKeys       []string
}

// Resources keeps track of AWS resources created during the ObjectLockScenario
 and handles
// cleanup when the scenario finishes.
type Resources struct {
 demoBuckets map[string]*DemoBucket

 s3Actions  *actions.S3Actions
 questioner demotools.IQuestioner
}

// init initializes objects in the Resources struct.
func (resources *Resources) init(s3Actions *actions.S3Actions, questioner
 demotools.IQuestioner) {
 resources.s3Actions = s3Actions
 resources.questioner = questioner
 resources.demoBuckets = map[string]*DemoBucket{}
}

// Cleanup deletes all AWS resources created during the ObjectLockScenario.
func (resources *Resources) Cleanup(ctx context.Context) {
 defer func() {
  if r := recover(); r != nil {
   log.Printf("Something went wrong during cleanup.\n%v\n", r)
   log.Println("Use the AWS Management Console to remove any remaining resources
 " +
    "that were created for this scenario.")
  }
 }()
```

```
 wantDelete := resources.questioner.AskBool("Do you want to remove all of the AWS
 resources that were created "+
  "during this demo (y/n)?", "y")
 if !wantDelete {
  log.Println("Be sure to remove resources when you're done with them to avoid
 unexpected charges!")
  return
 }

 log.Println("Removing objects from S3 buckets and deleting buckets...")
 resources.deleteBuckets(ctx)
 //resources.deleteRetentionObjects(resources.retentionBucket,
 resources.retentionObjects)

 log.Println("Cleanup complete.")
}

// deleteBuckets empties and then deletes all buckets created during the
 ObjectLockScenario.
func (resources *Resources) deleteBuckets(ctx context.Context) {
 for _, info := range createInfo {
  bucket := resources.demoBuckets[info.name]
  resources.deleteObjects(ctx, bucket)
  _, err := resources.s3Actions.S3Client.DeleteBucket(ctx, &s3.DeleteBucketInput{
   Bucket: aws.String(bucket.name),
  })
  if err != nil {
   panic(err)
  }
 }
 resources.demoBuckets = map[string]*DemoBucket{}
}

// deleteObjects deletes all objects in the specified bucket.
func (resources *Resources) deleteObjects(ctx context.Context, bucket
 *DemoBucket) {
 lockConfig, err := resources.s3Actions.GetObjectLockConfiguration(ctx,
 bucket.name)
 if err != nil {
  panic(err)
 }
 versions, err := resources.s3Actions.ListObjectVersions(ctx, bucket.name)
 if err != nil {
  switch err.(type) {
```

```
 case *types.NoSuchBucket:
  log.Printf("No objects to get from %s.\n", bucket.name)
 default:
  panic(err)
 }
}
delObjects := make([]types.ObjectIdentifier, len(versions))
for i, version := range versions {
 if lockConfig != nil && lockConfig.ObjectLockEnabled ==
types.ObjectLockEnabledEnabled {
  status, err := resources.s3Actions.GetObjectLegalHold(ctx, bucket.name,
*version.Key, *version.VersionId)
  if err != nil {
   switch err.(type) {
   case *types.NoSuchKey:
    log.Printf("Couldn't determine legal hold status for %s in %s.\n",
*version.Key, bucket.name)
   default:
    panic(err)
   }
  } else if status != nil && *status == types.ObjectLockLegalHoldStatusOn {
   err = resources.s3Actions.PutObjectLegalHold(ctx, bucket.name, *version.Key,
*version.VersionId, types.ObjectLockLegalHoldStatusOff)
   if err != nil {
    switch err.(type) {
    case *types.NoSuchKey:
     log.Printf("Couldn't turn off legal hold for %s in %s.\n", *version.Key,
bucket.name)
    default:
     panic(err)
    }
   }
  }
 }
 delObjects[i] = types.ObjectIdentifier{Key: version.Key, VersionId:
version.VersionId}
}
err = resources.s3Actions.DeleteObjects(ctx, bucket.name, delObjects,
bucket.retentionEnabled)
if err != nil {
 switch err.(type) {
 case *types.NoSuchBucket:
  log.Println("Nothing to delete.")
 default:
```

```
    panic(err)
   }
  }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Go .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif illustrant les fonctionnalités de verrouillage d'objets d'Amazon S3.

```
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import java.io.BufferedWriter;
import java.io.IOException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;
```

```
/*
 Before running this Java V2 code example, set up your development
 environment, including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/setup.html

 This Java example performs the following tasks:
    1. Create test Amazon Simple Storage Service (S3) buckets with different lock
 policies.
    2. Upload sample objects to each bucket.
    3. Set some Legal Hold and Retention Periods on objects and buckets.
    4. Investigate lock policies by viewing settings or attempting to delete or
 overwrite objects.
    5. Clean up objects and buckets.
 */
public class S3ObjectLockWorkflow {

    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    static String bucketName;
    static S3LockActions s3LockActions;
    private static final List<String> bucketNames = new ArrayList<>();
    private static final List<String> fileNames = new ArrayList<>();

    public static void main(String[] args) {
        // Get the current date and time to ensure bucket name is unique.
        LocalDateTime currentTime = LocalDateTime.now();

        // Format the date and time as a string.
        DateTimeFormatter formatter =
 DateTimeFormatter.ofPattern("yyyyMMddHHmmss");
        String timeStamp = currentTime.format(formatter);

        s3LockActions = new S3LockActions();
        bucketName = "bucket"+timeStamp;
        Scanner scanner = new Scanner(System.in);

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon Simple Storage Service (S3)
 Object Locking Workflow Scenario.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
```

```
        configurationSetup();
        System.out.println(DASHES);

        System.out.println(DASHES);
        setup();
        System.out.println("Setup is complete. Press Enter to continue...");
        scanner.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Lets present the user with choices.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        demoActionChoices() ;
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Would you like to clean up the resources? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            cleanup();
            System.out.println("Clean up is complete.");
        }

        System.out.println("Press Enter to continue...");
        scanner.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Amazon S3 Object Locking Workflow is complete.");
        System.out.println(DASHES);
    }

    // Present the user with the demo action choices.
    public static void demoActionChoices() {
        String[] choices = {
            "List all files in buckets.",
            "Attempt to delete a file.",
            "Attempt to delete a file with retention period bypass.",
            "Attempt to overwrite a file.",
            "View the object and bucket retention settings for a file.",
            "View the legal hold settings for a file.",
            "Finish the workflow."
        };
```

```
        int choice = 0;
        while (true) {
            System.out.println(DASHES);
            choice = getChoiceResponse("Explore the S3 locking features by
selecting one of the following choices:", choices);
            System.out.println(DASHES);
            System.out.println("You selected "+choices[choice]);
            switch (choice) {
                case 0 -> {
                    s3LockActions.listBucketsAndObjects(bucketNames, true);
                }

                case 1 -> {
                    System.out.println("Enter the number of the object to
delete:");
                    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                    List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                    String[] fileKeysArray = fileKeys.toArray(new String[0]);
                    int fileChoice = getChoiceResponse(null, fileKeysArray);
                    String objectKey = fileKeys.get(fileChoice);
                    String bucketName = allFiles.get(fileChoice).getBucketName();
                    String version = allFiles.get(fileChoice).getVersion();
                    s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
false, version);
                }

                case 2 -> {
                    System.out.println("Enter the number of the object to
delete:");
                    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                    List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                    String[] fileKeysArray = fileKeys.toArray(new String[0]);
                    int fileChoice = getChoiceResponse(null, fileKeysArray);
                    String objectKey = fileKeys.get(fileChoice);
                    String bucketName = allFiles.get(fileChoice).getBucketName();
                    String version = allFiles.get(fileChoice).getVersion();
                    s3LockActions.deleteObjectFromBucket(bucketName, objectKey,
true, version);
                }
```

```
                case 3 -> {
                    System.out.println("Enter the number of the object to
overwrite:");
                    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                    List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                    String[] fileKeysArray = fileKeys.toArray(new String[0]);
                    int fileChoice = getChoiceResponse(null, fileKeysArray);
                    String objectKey = fileKeys.get(fileChoice);
                    String bucketName = allFiles.get(fileChoice).getBucketName();

                    // Attempt to overwrite the file.
                    try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(objectKey))) {
                        writer.write("This is a modified text.");

                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    s3LockActions.uploadFile(bucketName, objectKey, objectKey);
                }

                case 4 -> {
                    System.out.println("Enter the number of the object to
overwrite:");
                    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
                    List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                    String[] fileKeysArray = fileKeys.toArray(new String[0]);
                    int fileChoice = getChoiceResponse(null, fileKeysArray);
                    String objectKey = fileKeys.get(fileChoice);
                    String bucketName = allFiles.get(fileChoice).getBucketName();
                    s3LockActions.getObjectRetention(bucketName, objectKey);
                }

                case 5 -> {
                    System.out.println("Enter the number of the object to
view:");
                    List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, true);
```

```
                    List<String> fileKeys = allFiles.stream().map(f ->
f.getKeyName()).collect(Collectors.toList());
                    String[] fileKeysArray = fileKeys.toArray(new String[0]);
                    int fileChoice = getChoiceResponse(null, fileKeysArray);
                    String objectKey = fileKeys.get(fileChoice);
                    String bucketName = allFiles.get(fileChoice).getBucketName();
                    s3LockActions.getObjectLegalHold(bucketName, objectKey);
                    s3LockActions.getBucketObjectLockConfiguration(bucketName);
                }

                case 6 -> {
                    System.out.println("Exiting the workflow...");
                    return;
                }

                default -> {
                    System.out.println("Invalid choice. Please select again.");
                }
            }
        }
    }

    // Clean up the resources from the scenario.
    private static void cleanup() {
        List<S3InfoObject> allFiles =
s3LockActions.listBucketsAndObjects(bucketNames, false);
        for (S3InfoObject fileInfo : allFiles) {
            String bucketName = fileInfo.getBucketName();
            String key = fileInfo.getKeyName();
            String version = fileInfo.getVersion();
            if (bucketName.contains("lock-enabled") ||
(bucketName.contains("retention-after-creation"))) {
                ObjectLockLegalHold legalHold =
s3LockActions.getObjectLegalHold(bucketName, key);
                if (legalHold != null) {
                    String holdStatus = legalHold.status().name();
                    System.out.println(holdStatus);
                    if (holdStatus.compareTo("ON") == 0) {
                        s3LockActions.modifyObjectLegalHold(bucketName, key,
false);
                    }
                }
                // Check for a retention period.
```

```
                ObjectLockRetention retention =
s3LockActions.getObjectRetention(bucketName, key);
                boolean hasRetentionPeriod ;
                hasRetentionPeriod = retention != null;
                s3LockActions.deleteObjectFromBucket(bucketName,
key,hasRetentionPeriod, version);

            } else {
                System.out.println(bucketName +" objects do not have a legal
lock");
                s3LockActions.deleteObjectFromBucket(bucketName, key,false,
version);
            }
        }

        // Delete the buckets.
        System.out.println("Delete "+bucketName);
        for (String bucket : bucketNames){
            s3LockActions.deleteBucketByName(bucket);
        }
    }

    private static void setup() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("""
                For this workflow, we will use the AWS SDK for Java to create
several S3
                buckets and files to demonstrate working with S3 locking
features.
                """);

        System.out.println("S3 buckets can be created either with or without
object lock enabled.");
        System.out.println("Press Enter to continue...");
        scanner.nextLine();

        // Create three S3 buckets.
        s3LockActions.createBucketWithLockOptions(false, bucketNames.get(0));
        s3LockActions.createBucketWithLockOptions(true, bucketNames.get(1));
        s3LockActions.createBucketWithLockOptions(false, bucketNames.get(2));
        System.out.println("Press Enter to continue.");
        scanner.nextLine();
```

```
        System.out.println("Bucket "+bucketNames.get(2) +" will be configured to
use object locking with a default retention period.");
        s3LockActions.modifyBucketDefaultRetention(bucketNames.get(2));
        System.out.println("Press Enter to continue.");
        scanner.nextLine();

        System.out.println("Object lock policies can also be added to existing
buckets. For this example, we will use "+bucketNames.get(1));
        s3LockActions.enableObjectLockOnBucket(bucketNames.get(1));
        System.out.println("Press Enter to continue.");
        scanner.nextLine();

        // Upload some files to the buckets.
        System.out.println("Now let's add some test files:");
        String fileName = "exampleFile.txt";
        int fileCount = 2;
        try (BufferedWriter writer = new BufferedWriter(new
java.io.FileWriter(fileName))) {
            writer.write("This is a sample file for uploading to a bucket.");

        } catch (IOException e) {
            e.printStackTrace();
        }

        for (String bucketName : bucketNames){
            for (int i = 0; i < fileCount; i++) {
                // Get the file name without extension.
                String fileNameWithoutExtension =
java.nio.file.Paths.get(fileName).getFileName().toString();
                int extensionIndex = fileNameWithoutExtension.lastIndexOf('.');
                if (extensionIndex > 0) {
                    fileNameWithoutExtension =
fileNameWithoutExtension.substring(0, extensionIndex);
                }

                // Create the numbered file names.
                String numberedFileName = fileNameWithoutExtension + i +
getFileExtension(fileName);
                fileNames.add(numberedFileName);
                s3LockActions.uploadFile(bucketName, numberedFileName, fileName);
            }
        }

        String question = null;
```

```
        System.out.print("Press Enter to continue...");
        scanner.nextLine();
        System.out.println("Now we can set some object lock policies on
individual files:");
        for (String bucketName : bucketNames) {
            for (int i = 0; i < fileNames.size(); i++){

                // No modifications to the objects in the first bucket.
                if (!bucketName.equals(bucketNames.get(0))) {
                    String exampleFileName = fileNames.get(i);
                    switch (i) {
                        case 0 -> {
                            question = "Would you like to add a legal hold to " +
exampleFileName + " in " + bucketName + " (y/n)?";
                            System.out.println(question);
                            String ans = scanner.nextLine().trim();
                            if (ans.equalsIgnoreCase("y")) {
                                System.out.println("**** You have selected to put
a legal hold " + exampleFileName);

                                // Set a legal hold.
                                s3LockActions.modifyObjectLegalHold(bucketName,
exampleFileName, true);
                            }
                        }
                        case 1 -> {
                            """
                                Would you like to add a 1 day Governance
retention period to %s in %s (y/n)?
                                Reminder: Only a user with the
s3:BypassGovernanceRetention permission will be able to delete this file or its
bucket until the retention period has expired.
                                """.formatted(exampleFileName, bucketName);
                            System.out.println(question);
                            String ans2 = scanner.nextLine().trim();
                            if (ans2.equalsIgnoreCase("y")) {

s3LockActions.modifyObjectRetentionPeriod(bucketName, exampleFileName);
                            }
                        }
                    }
                }
            }
        }
```

```
    }

    // Get file extension.
    private static String getFileExtension(String fileName) {
        int dotIndex = fileName.lastIndexOf('.');
        if (dotIndex > 0) {
            return fileName.substring(dotIndex);
        }
        return "";
    }

    public static void configurationSetup() {
        String noLockBucketName = bucketName + "-no-lock";
        String lockEnabledBucketName = bucketName + "-lock-enabled";
        String retentionAfterCreationBucketName = bucketName + "-retention-after-
creation";
        bucketNames.add(noLockBucketName);
        bucketNames.add(lockEnabledBucketName);
        bucketNames.add(retentionAfterCreationBucketName);
    }

    public static int getChoiceResponse(String question, String[] choices) {
        Scanner scanner = new Scanner(System.in);
        if (question != null) {
            System.out.println(question);
            for (int i = 0; i < choices.length; i++) {
                System.out.println("\t" + (i + 1) + ". " + choices[i]);
            }
        }

        int choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > choices.length) {
            String choice = scanner.nextLine();
            try {
                choiceNumber = Integer.parseInt(choice);
            } catch (NumberFormatException e) {
                System.out.println("Invalid choice. Please enter a valid
 number.");
            }
        }

        return choiceNumber - 1;
    }
}
```

Une classe wrapper pour les fonctions S3.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketVersioningStatus;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.CreateBucketRequest;
import software.amazon.awssdk.services.s3.model.DefaultRetention;
import software.amazon.awssdk.services.s3.model.DeleteBucketRequest;
import software.amazon.awssdk.services.s3.model.DeleteObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldRequest;
import software.amazon.awssdk.services.s3.model.GetObjectLegalHoldResponse;
import
 software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationRequest;
import
 software.amazon.awssdk.services.s3.model.GetObjectLockConfigurationResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.GetObjectRetentionResponse;
import software.amazon.awssdk.services.s3.model.HeadBucketRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsRequest;
import software.amazon.awssdk.services.s3.model.ListObjectVersionsResponse;
import software.amazon.awssdk.services.s3.model.MFADelete;
import software.amazon.awssdk.services.s3.model.ObjectLockConfiguration;
import software.amazon.awssdk.services.s3.model.ObjectLockEnabled;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHold;
import software.amazon.awssdk.services.s3.model.ObjectLockLegalHoldStatus;
import software.amazon.awssdk.services.s3.model.ObjectLockRetention;
import software.amazon.awssdk.services.s3.model.ObjectLockRetentionMode;
import software.amazon.awssdk.services.s3.model.ObjectLockRule;
import software.amazon.awssdk.services.s3.model.PutBucketVersioningRequest;
import software.amazon.awssdk.services.s3.model.PutObjectLegalHoldRequest;
import
 software.amazon.awssdk.services.s3.model.PutObjectLockConfigurationRequest;
import software.amazon.awssdk.services.s3.model.PutObjectRequest;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.PutObjectRetentionRequest;
import software.amazon.awssdk.services.s3.model.S3Exception;
import software.amazon.awssdk.services.s3.model.VersioningConfiguration;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import java.nio.file.Path;
import java.nio.file.Paths;
```

```
import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.List;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.stream.Collectors;

// Contains application logic for the Amazon S3 operations used in this workflow.
public class S3LockActions {

    private static S3Client getClient() {
        return S3Client.builder()
            .region(Region.US_EAST_1)
            .build();
    }

    // Set or modify a retention period on an object in an S3 bucket.
    public void modifyObjectRetentionPeriod(String bucketName, String objectKey)
 {
        // Calculate the instant one day from now.
        Instant futureInstant = Instant.now().plus(1, ChronoUnit.DAYS);

        // Convert the Instant to a ZonedDateTime object with a specific time
 zone.
        ZonedDateTime zonedDateTime =
 futureInstant.atZone(ZoneId.systemDefault());

        // Define a formatter for human-readable output.
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
 HH:mm:ss");

        // Format the ZonedDateTime object to a human-readable date string.
        String humanReadableDate = formatter.format(zonedDateTime);

        // Print the formatted date string.
        System.out.println("Formatted Date: " + humanReadableDate);
        ObjectLockRetention retention = ObjectLockRetention.builder()
            .mode(ObjectLockRetentionMode.GOVERNANCE)
            .retainUntilDate(futureInstant)
            .build();
```

```
        PutObjectRetentionRequest retentionRequest =
PutObjectRetentionRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .retention(retention)
            .build();

        getClient().putObjectRetention(retentionRequest);
        System.out.println("Set retention for "+objectKey +" in " +bucketName +"
until "+ humanReadableDate +".");
    }

    // Get the legal hold details for an S3 object.
    public ObjectLockLegalHold getObjectLegalHold(String bucketName, String
objectKey) {
        try {
            GetObjectLegalHoldRequest legalHoldRequest =
GetObjectLegalHoldRequest.builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();

            GetObjectLegalHoldResponse response =
getClient().getObjectLegalHold(legalHoldRequest);
            System.out.println("Object legal hold for " + objectKey + " in " +
bucketName +
                ":\n\tStatus: " + response.legalHold().status());
            return response.legalHold();

        } catch (S3Exception ex) {
            System.out.println("\tUnable to fetch legal hold: '" +
ex.getMessage() + "'");
        }

        return null;
    }

    // Create a new Amazon S3 bucket with object lock options.
    public void createBucketWithLockOptions(boolean enableObjectLock, String
bucketName) {
        S3Waiter s3Waiter = getClient().waiter();
        CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectLockEnabledForBucket(enableObjectLock)
```

```
            .build();

        getClient().createBucket(bucketRequest);
        HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
            .bucket(bucketName)
            .build();

        // Wait until the bucket is created and print out the response.
        s3Waiter.waitUntilBucketExists(bucketRequestWait);
        System.out.println(bucketName + " is ready");
    }

    public List<S3InfoObject> listBucketsAndObjects(List<String> bucketNames,
Boolean interactive) {
        AtomicInteger counter = new AtomicInteger(0); // Initialize counter.
        return bucketNames.stream()
            .flatMap(bucketName ->
listBucketObjectsAndVersions(bucketName).versions().stream()
                .map(version -> {
                    S3InfoObject s3InfoObject = new S3InfoObject();
                    s3InfoObject.setBucketName(bucketName);
                    s3InfoObject.setVersion(version.versionId());
                    s3InfoObject.setKeyName(version.key());
                    return s3InfoObject;
                }))
            .peek(s3InfoObject -> {
                int i = counter.incrementAndGet(); // Increment and get the
updated value.
                if (interactive) {
                    System.out.println(i + ": "+ s3InfoObject.getKeyName());
                    System.out.printf("%5s Bucket name: %s\n", "",
s3InfoObject.getBucketName());
                    System.out.printf("%5s Version: %s\n", "",
s3InfoObject.getVersion());
                }
            })
            .collect(Collectors.toList());
    }

    public ListObjectVersionsResponse listBucketObjectsAndVersions(String
bucketName) {
        ListObjectVersionsRequest versionsRequest =
ListObjectVersionsRequest.builder()
            .bucket(bucketName)
```

```
            .build();

        return getClient().listObjectVersions(versionsRequest);
    }

    // Set or modify a retention period on an S3 bucket.
    public void modifyBucketDefaultRetention(String bucketName) {
        VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
            .mfaDelete(MFADelete.DISABLED)
            .status(BucketVersioningStatus.ENABLED)
            .build();

        PutBucketVersioningRequest versioningRequest =
PutBucketVersioningRequest.builder()
            .bucket(bucketName)
            .versioningConfiguration(versioningConfiguration)
            .build();

        getClient().putBucketVersioning(versioningRequest);
        DefaultRetention rention = DefaultRetention.builder()
            .days(1)
            .mode(ObjectLockRetentionMode.GOVERNANCE)
            .build();

        ObjectLockRule lockRule = ObjectLockRule.builder()
            .defaultRetention(rention)
            .build();

        ObjectLockConfiguration objectLockConfiguration =
ObjectLockConfiguration.builder()
            .objectLockEnabled(ObjectLockEnabled.ENABLED)
            .rule(lockRule)
            .build();

        PutObjectLockConfigurationRequest putObjectLockConfigurationRequest =
PutObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .objectLockConfiguration(objectLockConfiguration)
            .build();


getClient().putObjectLockConfiguration(putObjectLockConfigurationRequest) ;
```

```
        System.out.println("Added a default retention to bucket "+bucketName
+".");
    }

    // Enable object lock on an existing bucket.
    public void enableObjectLockOnBucket(String bucketName) {
        try {
            VersioningConfiguration versioningConfiguration =
VersioningConfiguration.builder()
                .status(BucketVersioningStatus.ENABLED)
                .build();

            PutBucketVersioningRequest putBucketVersioningRequest =
PutBucketVersioningRequest.builder()
                .bucket(bucketName)
                .versioningConfiguration(versioningConfiguration)
                .build();

            // Enable versioning on the bucket.
            getClient().putBucketVersioning(putBucketVersioningRequest);
            PutObjectLockConfigurationRequest request =
PutObjectLockConfigurationRequest.builder()
                .bucket(bucketName)
                .objectLockConfiguration(ObjectLockConfiguration.builder()
                    .objectLockEnabled(ObjectLockEnabled.ENABLED)
                    .build())
                .build();

            getClient().putObjectLockConfiguration(request);
            System.out.println("Successfully enabled object lock on
"+bucketName);

        } catch (S3Exception ex) {
            System.out.println("Error modifying object lock: '" + ex.getMessage()
+ "'");
        }
    }

    public void uploadFile(String bucketName, String objectName, String filePath)
{
        Path file = Paths.get(filePath);
        PutObjectRequest request = PutObjectRequest.builder()
            .bucket(bucketName)
            .key(objectName)
```

```
            .checksumAlgorithm(ChecksumAlgorithm.SHA256)
            .build();

        PutObjectResponse response = getClient().putObject(request, file);
        if (response != null) {
            System.out.println("\tSuccessfully uploaded " + objectName + " to " +
bucketName + ".");
        } else {
            System.out.println("\tCould not upload " + objectName + " to " +
bucketName + ".");
        }
    }

    // Set or modify a legal hold on an object in an S3 bucket.
    public void modifyObjectLegalHold(String bucketName, String objectKey,
boolean legalHoldOn) {
        ObjectLockLegalHold legalHold ;
        if (legalHoldOn) {
            legalHold = ObjectLockLegalHold.builder()
                .status(ObjectLockLegalHoldStatus.ON)
                .build();
        } else {
            legalHold = ObjectLockLegalHold.builder()
                .status(ObjectLockLegalHoldStatus.OFF)
                .build();
        }

        PutObjectLegalHoldRequest legalHoldRequest =
PutObjectLegalHoldRequest.builder()
            .bucket(bucketName)
            .key(objectKey)
            .legalHold(legalHold)
            .build();

        getClient().putObjectLegalHold(legalHoldRequest) ;
        System.out.println("Modified legal hold for "+ objectKey +" in
"+bucketName +".");
    }

    // Delete an object from a specific bucket.
    public void deleteObjectFromBucket(String bucketName, String objectKey,
boolean hasRetention, String versionId) {
        try {
            DeleteObjectRequest objectRequest;
```

```
            if (hasRetention) {
                objectRequest = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .versionId(versionId)
                    .bypassGovernanceRetention(true)
                    .build();
            } else {
                objectRequest = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .versionId(versionId)
                    .build();
            }

            getClient().deleteObject(objectRequest) ;
            System.out.println("The object was successfully deleted");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    // Get the retention period for an S3 object.
    public ObjectLockRetention getObjectRetention(String bucketName, String key){
        try {
            GetObjectRetentionRequest retentionRequest =
GetObjectRetentionRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();

            GetObjectRetentionResponse response =
getClient().getObjectRetention(retentionRequest);
            System.out.println("tObject retention for "+key +"
in "+ bucketName +": " + response.retention().mode() +" until "+
response.retention().retainUntilDate() +".");
            return response.retention();

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return null;
        }
    }
```

```
    public void deleteBucketByName(String bucketName) {
        try {
            DeleteBucketRequest request = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();

            getClient().deleteBucket(request);
            System.out.println(bucketName +" was deleted.");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
    }

    // Get the object lock configuration details for an S3 bucket.
    public void getBucketObjectLockConfiguration(String bucketName) {
        GetObjectLockConfigurationRequest objectLockConfigurationRequest =
 GetObjectLockConfigurationRequest.builder()
            .bucket(bucketName)
            .build();

        GetObjectLockConfigurationResponse response =
 getClient().getObjectLockConfiguration(objectLockConfigurationRequest);
        System.out.println("Bucket object lock config for "+bucketName +":  ");
        System.out.println("\tEnabled:
 "+response.objectLockConfiguration().objectLockEnabled());
        System.out.println("\tRule: "+
 response.objectLockConfiguration().rule().defaultRetention());
    }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

index.js- Point d'entrée pour le flux de travail. Cela permet d'orchestrer toutes les étapes. Consultez GitHub les détails de mise en œuvre de Scenario ScenarioInput, ScenarioOutput, et ScenarioAction.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import * as Scenarios from "@aws-doc-sdk-examples/lib/scenario/index.js";
import {
  exitOnFalse,
  loadState,
  saveState,
} from "@aws-doc-sdk-examples/lib/scenario/steps-common.js";

import { welcome, welcomeContinue } from "./welcome.steps.js";
import {
  confirmCreateBuckets,
  confirmPopulateBuckets,
  confirmSetLegalHoldFileEnabled,
  confirmSetLegalHoldFileRetention,
  confirmSetRetentionPeriodFileEnabled,
  confirmSetRetentionPeriodFileRetention,
  confirmUpdateLockPolicy,
  confirmUpdateRetention,
  createBuckets,
  createBucketsAction,
  populateBuckets,
  populateBucketsAction,
  setLegalHoldFileEnabledAction,
  setLegalHoldFileRetentionAction,
  setRetentionPeriodFileEnabledAction,
  setRetentionPeriodFileRetentionAction,
  updateLockPolicy,
```

```
  updateLockPolicyAction,
  updateRetention,
  updateRetentionAction,
} from "./setup.steps.js";

/**
 * @param {Scenarios} scenarios
 * @param {Record<string, any>} initialState
 */
export const getWorkflowStages = (scenarios, initialState = {}) => {
  const client = new S3Client({});

  return {
    deploy: new scenarios.Scenario(
      "S3 Object Locking - Deploy",
      [
        welcome(scenarios),
        welcomeContinue(scenarios),
        exitOnFalse(scenarios, "welcomeContinue"),
        createBuckets(scenarios),
        confirmCreateBuckets(scenarios),
        exitOnFalse(scenarios, "confirmCreateBuckets"),
        createBucketsAction(scenarios, client),
        updateRetention(scenarios),
        confirmUpdateRetention(scenarios),
        exitOnFalse(scenarios, "confirmUpdateRetention"),
        updateRetentionAction(scenarios, client),
        populateBuckets(scenarios),
        confirmPopulateBuckets(scenarios),
        exitOnFalse(scenarios, "confirmPopulateBuckets"),
        populateBucketsAction(scenarios, client),
        updateLockPolicy(scenarios),
        confirmUpdateLockPolicy(scenarios),
        exitOnFalse(scenarios, "confirmUpdateLockPolicy"),
        updateLockPolicyAction(scenarios, client),
        confirmSetLegalHoldFileEnabled(scenarios),
        setLegalHoldFileEnabledAction(scenarios, client),
        confirmSetRetentionPeriodFileEnabled(scenarios),
        setRetentionPeriodFileEnabledAction(scenarios, client),
        confirmSetLegalHoldFileRetention(scenarios),
        setLegalHoldFileRetentionAction(scenarios, client),
        confirmSetRetentionPeriodFileRetention(scenarios),
        setRetentionPeriodFileRetentionAction(scenarios, client),
        saveState,
```

```
      ],
      initialState,
    ),
    demo: new scenarios.Scenario(
      "S3 Object Locking - Demo",
      [loadState, replAction(scenarios, client)],
      initialState,
    ),
    clean: new scenarios.Scenario(
      "S3 Object Locking - Destroy",
      [
        loadState,
        confirmCleanup(scenarios),
        exitOnFalse(scenarios, "confirmCleanup"),
        cleanupAction(scenarios, client),
      ],
      initialState,
    ),
  };
};

// Call function if run directly
import { fileURLToPath } from "url";
import { S3Client } from "@aws-sdk/client-s3";
import { cleanupAction, confirmCleanup } from "./clean.steps.js";
import { replAction } from "./repl.steps.js";

if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const objectLockingScenarios = getWorkflowStages(Scenarios);
  Scenarios.parseScenarioArgs(objectLockingScenarios);
}
```

welcome.steps.js- Envoyez des messages de bienvenue sur la console.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
 * @param {Scenarios} scenarios
```

```
 */
const welcome = (scenarios) =>
  new scenarios.ScenarioOutput(
    "welcome",
    `Welcome to the Amazon Simple Storage Service (S3) Object Locking Workflow
 Scenario. For this workflow, we will use the AWS SDK for JavaScript to create
 several S3 buckets and files to demonstrate working with S3 locking features.`,
    { header: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const welcomeContinue = (scenarios) =>
  new scenarios.ScenarioInput(
    "welcomeContinue",
    "Press Enter when you are ready to start.",
    { type: "confirm" },
  );

export { welcome, welcomeContinue };
```

setup.steps.js- Déployez des compartiments, des objets et des paramètres de fichier.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  BucketVersioningStatus,
  ChecksumAlgorithm,
  CreateBucketCommand,
  MFADeleteStatus,
  PutBucketVersioningCommand,
  PutObjectCommand,
  PutObjectLockConfigurationCommand,
  PutObjectLegalHoldCommand,
  PutObjectRetentionCommand,
  ObjectLockLegalHoldStatus,
  ObjectLockRetentionMode,
} from "@aws-sdk/client-s3";

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
```

```
 */

/**
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client
 */

const bucketPrefix = "js-object-locking";

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const createBuckets = (scenarios) =>
  new scenarios.ScenarioOutput(
    "createBuckets",
    `The following buckets will be created:
         ${bucketPrefix}-no-lock with object lock False.
         ${bucketPrefix}-lock-enabled with object lock True.
         ${bucketPrefix}-retention-after-creation with object lock False.`,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmCreateBuckets = (scenarios) =>
  new scenarios.ScenarioInput("confirmCreateBuckets", "Create the buckets?", {
    type: "confirm",
  });

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const createBucketsAction = (scenarios, client) =>
  new scenarios.ScenarioAction("createBucketsAction", async (state) => {
    const noLockBucketName = `${bucketPrefix}-no-lock`;
    const lockEnabledBucketName = `${bucketPrefix}-lock-enabled`;
    const retentionBucketName = `${bucketPrefix}-retention-after-creation`;

    await client.send(new CreateBucketCommand({ Bucket: noLockBucketName }));
    await client.send(
      new CreateBucketCommand({
        Bucket: lockEnabledBucketName,
```

```
        ObjectLockEnabledForBucket: true,
      }),
    );
    await client.send(new CreateBucketCommand({ Bucket: retentionBucketName }));

    state.noLockBucketName = noLockBucketName;
    state.lockEnabledBucketName = lockEnabledBucketName;
    state.retentionBucketName = retentionBucketName;
  });

/**
 * @param {Scenarios} scenarios
 */
const populateBuckets = (scenarios) =>
  new scenarios.ScenarioOutput(
    "populateBuckets",
    `The following test files will be created:
         file0.txt in ${bucketPrefix}-no-lock.
         file1.txt in ${bucketPrefix}-no-lock.
         file0.txt in ${bucketPrefix}-lock-enabled.
         file1.txt in ${bucketPrefix}-lock-enabled.
         file0.txt in ${bucketPrefix}-retention-after-creation.
         file1.txt in ${bucketPrefix}-retention-after-creation.`,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmPopulateBuckets = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmPopulateBuckets",
    "Populate the buckets?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const populateBucketsAction = (scenarios, client) =>
  new scenarios.ScenarioAction("populateBucketsAction", async (state) => {
    await client.send(
      new PutObjectCommand({
```

```
        Bucket: state.noLockBucketName,
        Key: "file0.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
      }),
    );
    await client.send(
      new PutObjectCommand({
        Bucket: state.noLockBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
      }),
    );
    await client.send(
      new PutObjectCommand({
        Bucket: state.lockEnabledBucketName,
        Key: "file0.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
      }),
    );
    await client.send(
      new PutObjectCommand({
        Bucket: state.lockEnabledBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
      }),
    );
    await client.send(
      new PutObjectCommand({
        Bucket: state.retentionBucketName,
        Key: "file0.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
      }),
    );
    await client.send(
      new PutObjectCommand({
        Bucket: state.retentionBucketName,
        Key: "file1.txt",
        Body: "Content",
        ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
```

```
      }),
    );
  });

/**
 * @param {Scenarios} scenarios
 */
const updateRetention = (scenarios) =>
  new scenarios.ScenarioOutput(
    "updateRetention",
    `A bucket can be configured to use object locking with a default retention
 period.
   A default retention period will be configured for ${bucketPrefix}-retention-
after-creation.`,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmUpdateRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmUpdateRetention",
    "Configure default retention period?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const updateRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction("updateRetentionAction", async (state) => {
    await client.send(
      new PutBucketVersioningCommand({
        Bucket: state.retentionBucketName,
        VersioningConfiguration: {
          MFADelete: MFADeleteStatus.Disabled,
          Status: BucketVersioningStatus.Enabled,
        },
      }),
    );

    await client.send(
```

```
      new PutObjectLockConfigurationCommand({
        Bucket: state.retentionBucketName,
        ObjectLockConfiguration: {
          ObjectLockEnabled: "Enabled",
          Rule: {
            DefaultRetention: {
              Mode: "GOVERNANCE",
              Years: 1,
            },
          },
        },
      }),
    );
  });

/**
 * @param {Scenarios} scenarios
 */
const updateLockPolicy = (scenarios) =>
  new scenarios.ScenarioOutput(
    "updateLockPolicy",
    `Object lock policies can also be added to existing buckets.
   An object lock policy will be added to ${bucketPrefix}-lock-enabled.`,
    { preformatted: true },
  );

/**
 * @param {Scenarios} scenarios
 */
const confirmUpdateLockPolicy = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmUpdateLockPolicy",
    "Add object lock policy?",
    { type: "confirm" },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const updateLockPolicyAction = (scenarios, client) =>
  new scenarios.ScenarioAction("updateLockPolicyAction", async (state) => {
    await client.send(
      new PutObjectLockConfigurationCommand({
```

```
        Bucket: state.lockEnabledBucketName,
        ObjectLockConfiguration: {
          ObjectLockEnabled: "Enabled",
        },
      }),
    );
  });

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetLegalHoldFileEnabled = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetLegalHoldFileEnabled",
    (state) =>
      `Would you like to add a legal hold to file0.txt in
 ${state.lockEnabledBucketName}?`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setLegalHoldFileEnabledAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setLegalHoldFileEnabledAction",
    async (state) => {
      await client.send(
        new PutObjectLegalHoldCommand({
          Bucket: state.lockEnabledBucketName,
          Key: "file0.txt",
          LegalHold: {
            Status: ObjectLockLegalHoldStatus.ON,
          },
        }),
      );
      console.log(
        `Modified legal hold for file0.txt in ${state.lockEnabledBucketName}.`,
      );
    },
```

```
    { skipWhen: (state) => !state.confirmSetLegalHoldFileEnabled },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetRetentionPeriodFileEnabled = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetRetentionPeriodFileEnabled",
    (state) =>
      `Would you like to add a 1 day Governance retention period to file1.txt in
 ${state.lockEnabledBucketName}?
Reminder: Only a user with the s3:BypassGovernanceRetention permission will be
 able to delete this file or its bucket until the retention period has expired.`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setRetentionPeriodFileEnabledAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setRetentionPeriodFileEnabledAction",
    async (state) => {
      const retentionDate = new Date();
      retentionDate.setDate(retentionDate.getDate() + 1);
      await client.send(
        new PutObjectRetentionCommand({
          Bucket: state.lockEnabledBucketName,
          Key: "file1.txt",
          Retention: {
            Mode: ObjectLockRetentionMode.GOVERNANCE,
            RetainUntilDate: retentionDate,
          },
        }),
      );
      console.log(
        `Set retention for file1.txt in ${state.lockEnabledBucketName} until
 ${retentionDate.toISOString().split("T")[0]}.`,
      );
```

```
    },
    { skipWhen: (state) => !state.confirmSetRetentionPeriodFileEnabled },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const confirmSetLegalHoldFileRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetLegalHoldFileRetention",
    (state) =>
      `Would you like to add a legal hold to file0.txt in
 ${state.retentionBucketName}?`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setLegalHoldFileRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setLegalHoldFileRetentionAction",
    async (state) => {
      await client.send(
        new PutObjectLegalHoldCommand({
          Bucket: state.retentionBucketName,
          Key: "file0.txt",
          LegalHold: {
            Status: ObjectLockLegalHoldStatus.ON,
          },
        }),
      );
      console.log(
        `Modified legal hold for file0.txt in ${state.retentionBucketName}.`,
      );
    },
    { skipWhen: (state) => !state.confirmSetLegalHoldFileRetention },
  );

/**
```

```
 * @param {Scenarios} scenarios
 */
const confirmSetRetentionPeriodFileRetention = (scenarios) =>
  new scenarios.ScenarioInput(
    "confirmSetRetentionPeriodFileRetention",
    (state) =>
      `Would you like to add a 1 day Governance retention period to file1.txt in
 ${state.retentionBucketName}?
Reminder: Only a user with the s3:BypassGovernanceRetention permission will be
 able to delete this file or its bucket until the retention period has expired.`,
    {
      type: "confirm",
    },
  );

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const setRetentionPeriodFileRetentionAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "setRetentionPeriodFileRetentionAction",
    async (state) => {
      const retentionDate = new Date();
      retentionDate.setDate(retentionDate.getDate() + 1);
      await client.send(
        new PutObjectRetentionCommand({
          Bucket: state.retentionBucketName,
          Key: "file1.txt",
          Retention: {
            Mode: ObjectLockRetentionMode.GOVERNANCE,
            RetainUntilDate: retentionDate,
          },
          BypassGovernanceRetention: true,
        }),
      );
      console.log(
        `Set retention for file1.txt in ${state.retentionBucketName} until
 ${retentionDate.toISOString().split("T")[0]}.`,
      );
    },
    { skipWhen: (state) => !state.confirmSetRetentionPeriodFileRetention },
  );
```

```
export {
  createBuckets,
  confirmCreateBuckets,
  createBucketsAction,
  populateBuckets,
  confirmPopulateBuckets,
  populateBucketsAction,
  updateRetention,
  confirmUpdateRetention,
  updateRetentionAction,
  updateLockPolicy,
  confirmUpdateLockPolicy,
  updateLockPolicyAction,
  confirmSetLegalHoldFileEnabled,
  setLegalHoldFileEnabledAction,
  confirmSetRetentionPeriodFileEnabled,
  setRetentionPeriodFileEnabledAction,
  confirmSetLegalHoldFileRetention,
  setLegalHoldFileRetentionAction,
  confirmSetRetentionPeriodFileRetention,
  setRetentionPeriodFileRetentionAction,
};
```

repl.steps.js- Afficher et supprimer des fichiers dans les compartiments.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  ChecksumAlgorithm,
  DeleteObjectCommand,
  GetObjectLegalHoldCommand,
  GetObjectLockConfigurationCommand,
  GetObjectRetentionCommand,
  ListObjectVersionsCommand,
  PutObjectCommand,
} from "@aws-sdk/client-s3";

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
```

```
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client
 */

const choices = {
  EXIT: 0,
  LIST_ALL_FILES: 1,
  DELETE_FILE: 2,
  DELETE_FILE_WITH_RETENTION: 3,
  OVERWRITE_FILE: 4,
  VIEW_RETENTION_SETTINGS: 5,
  VIEW_LEGAL_HOLD_SETTINGS: 6,
};

/**
 * @param {Scenarios} scenarios
 */
const replInput = (scenarios) =>
  new scenarios.ScenarioInput(
    "replChoice",
    `Explore the S3 locking features by selecting one of the following choices`,
    {
      type: "select",
      choices: [
        { name: "List all files in buckets", value: choices.LIST_ALL_FILES },
        { name: "Attempt to delete a file.", value: choices.DELETE_FILE },
        {
          name: "Attempt to delete a file with retention period bypass.",
          value: choices.DELETE_FILE_WITH_RETENTION,
        },
        { name: "Attempt to overwrite a file.", value: choices.OVERWRITE_FILE },
        {
          name: "View the object and bucket retention settings for a file.",
          value: choices.VIEW_RETENTION_SETTINGS,
        },
        {
          name: "View the legal hold settings for a file.",
          value: choices.VIEW_LEGAL_HOLD_SETTINGS,
        },
        { name: "Finish the workflow.", value: choices.EXIT },
      ],
    },
  );

/**
```

```
 * @param {S3Client} client
 * @param {string[]} buckets
 */
const getAllFiles = async (client, buckets) => {
  /** @type {{bucket: string, key: string, version: string}[]} */
  const files = [];
  for (const bucket of buckets) {
    const objectsResponse = await client.send(
      new ListObjectVersionsCommand({ Bucket: bucket }),
    );
    for (const version of objectsResponse.Versions || []) {
      const { Key, VersionId } = version;
      files.push({ bucket, key: Key, version: VersionId });
    }
  }

  return files;
};

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const replAction = (scenarios, client) =>
  new scenarios.ScenarioAction(
    "replAction",
    async (state) => {
      const files = await getAllFiles(client, [
        state.noLockBucketName,
        state.lockEnabledBucketName,
        state.retentionBucketName,
      ]);

      const fileInput = new scenarios.ScenarioInput(
        "selectedFile",
        "Select a file:",
        {
          type: "select",
          choices: files.map((file, index) => ({
            name: `${index + 1}: ${file.bucket}: ${file.key} (version: ${
              file.version
            })`,
            value: index,
          })),
```

```
        },
      );

      const { replChoice } = state;

      switch (replChoice) {
        case choices.LIST_ALL_FILES: {
          const files = await getAllFiles(client, [
            state.noLockBucketName,
            state.lockEnabledBucketName,
            state.retentionBucketName,
          ]);
          state.replOutput = files
            .map(
              (file) =>
                `${file.bucket}: ${file.key} (version: ${file.version})`,
            )
            .join("\n");
          break;
        }
        case choices.DELETE_FILE: {
          /** @type {number} */
          const fileToDelete = await fileInput.handle(state);
          const selectedFile = files[fileToDelete];
          try {
            await client.send(
              new DeleteObjectCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                VersionId: selectedFile.version,
              }),
            );
            state.replOutput = `Deleted ${selectedFile.key} in
${selectedFile.bucket}.`;
          } catch (err) {
            state.replOutput = `Unable to delete object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
          }
          break;
        }
        case choices.DELETE_FILE_WITH_RETENTION: {
          /** @type {number} */
          const fileToDelete = await fileInput.handle(state);
          const selectedFile = files[fileToDelete];
```

```
          try {
            await client.send(
              new DeleteObjectCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                VersionId: selectedFile.version,
                BypassGovernanceRetention: true,
              }),
            );
            state.replOutput = `Deleted ${selectedFile.key} in
${selectedFile.bucket}.`;
          } catch (err) {
            state.replOutput = `Unable to delete object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
          }
          break;
        }
        case choices.OVERWRITE_FILE: {
          /** @type {number} */
          const fileToOverwrite = await fileInput.handle(state);
          const selectedFile = files[fileToOverwrite];
          try {
            await client.send(
              new PutObjectCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                Body: "New content",
                ChecksumAlgorithm: ChecksumAlgorithm.SHA256,
              }),
            );
            state.replOutput = `Overwrote ${selectedFile.key} in
${selectedFile.bucket}.`;
          } catch (err) {
            state.replOutput = `Unable to overwrite object ${selectedFile.key} in
bucket ${selectedFile.bucket}: ${err.message}`;
          }
          break;
        }
        case choices.VIEW_RETENTION_SETTINGS: {
          /** @type {number} */
          const fileToView = await fileInput.handle(state);
          const selectedFile = files[fileToView];
          try {
            const retention = await client.send(
```

```
              new GetObjectRetentionCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                VersionId: selectedFile.version,
              }),
            );
            const bucketConfig = await client.send(
              new GetObjectLockConfigurationCommand({
                Bucket: selectedFile.bucket,
              }),
            );
            state.replOutput = `Object retention for ${selectedFile.key}
 in ${selectedFile.bucket}: ${retention.Retention?.Mode} until
 ${retention.Retention?.RetainUntilDate?.toISOString()}.
Bucket object lock config for ${selectedFile.bucket} in ${selectedFile.bucket}:
Enabled: ${bucketConfig.ObjectLockConfiguration?.ObjectLockEnabled}
Rule:
 ${JSON.stringify(bucketConfig.ObjectLockConfiguration?.Rule?.DefaultRetention)}`;
          } catch (err) {
            state.replOutput = `Unable to fetch object lock retention:
 '${err.message}'`;
          }
          break;
        }
        case choices.VIEW_LEGAL_HOLD_SETTINGS: {
          /** @type {number} */
          const fileToView = await fileInput.handle(state);
          const selectedFile = files[fileToView];
          try {
            const legalHold = await client.send(
              new GetObjectLegalHoldCommand({
                Bucket: selectedFile.bucket,
                Key: selectedFile.key,
                VersionId: selectedFile.version,
              }),
            );
            state.replOutput = `Object legal hold for ${selectedFile.key} in
 ${selectedFile.bucket}: Status: ${legalHold.LegalHold?.Status}`;
          } catch (err) {
            state.replOutput = `Unable to fetch legal hold: '${err.message}'`;
          }
          break;
        }
        default:
```

```
          throw new Error(`Invalid replChoice: ${replChoice}`);
      }
    },
    {
      whileConfig: {
        whileFn: ({ replChoice }) => replChoice !== choices.EXIT,
        input: replInput(scenarios),
        output: new scenarios.ScenarioOutput(
          "REPL output",
          (state) => state.replOutput,
          { preformatted: true },
        ),
      },
    },
  );

export { replInput, replAction, choices };
```

clean.steps.js- Détruisez toutes les ressources créées.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import {
  DeleteObjectCommand,
  DeleteBucketCommand,
  ListObjectVersionsCommand,
  GetObjectLegalHoldCommand,
  GetObjectRetentionCommand,
  PutObjectLegalHoldCommand,
} from "@aws-sdk/client-s3";

/**
 * @typedef {import("@aws-doc-sdk-examples/lib/scenario/index.js")} Scenarios
 */

/**
 * @typedef {import("@aws-sdk/client-s3").S3Client} S3Client
 */

/**
 * @param {Scenarios} scenarios
 */
```

```
const confirmCleanup = (scenarios) =>
  new scenarios.ScenarioInput("confirmCleanup", "Clean up resources?", {
    type: "confirm",
  });

/**
 * @param {Scenarios} scenarios
 * @param {S3Client} client
 */
const cleanupAction = (scenarios, client) =>
  new scenarios.ScenarioAction("cleanupAction", async (state) => {
    const { noLockBucketName, lockEnabledBucketName, retentionBucketName } =
      state;

    const buckets = [
      noLockBucketName,
      lockEnabledBucketName,
      retentionBucketName,
    ];

    for (const bucket of buckets) {
      /** @type {import("@aws-sdk/client-s3").ListObjectVersionsCommandOutput} */
      let objectsResponse;

      try {
        objectsResponse = await client.send(
          new ListObjectVersionsCommand({
            Bucket: bucket,
          }),
        );
      } catch (e) {
        if (e instanceof Error && e.name === "NoSuchBucket") {
          console.log("Object's bucket has already been deleted.");
          continue;
        } else {
          throw e;
        }
      }

      for (const version of objectsResponse.Versions || []) {
        const { Key, VersionId } = version;

        try {
          const legalHold = await client.send(
```

```
            new GetObjectLegalHoldCommand({
              Bucket: bucket,
              Key,
              VersionId,
            }),
          );

          if (legalHold.LegalHold?.Status === "ON") {
            await client.send(
              new PutObjectLegalHoldCommand({
                Bucket: bucket,
                Key,
                VersionId,
                LegalHold: {
                  Status: "OFF",
                },
              }),
            );
          }
        } catch (err) {
          console.log(
            `Unable to fetch legal hold for ${Key} in ${bucket}:
'${err.message}'`,
          );
        }

        try {
          const retention = await client.send(
            new GetObjectRetentionCommand({
              Bucket: bucket,
              Key,
              VersionId,
            }),
          );

          if (retention.Retention?.Mode === "GOVERNANCE") {
            await client.send(
              new DeleteObjectCommand({
                Bucket: bucket,
                Key,
                VersionId,
                BypassGovernanceRetention: true,
              }),
            );
```

```
          }
        } catch (err) {
          console.log(
            `Unable to fetch object lock retention for ${Key} in ${bucket}:
 '${err.message}'`,
          );
        }

        await client.send(
          new DeleteObjectCommand({
            Bucket: bucket,
            Key,
            VersionId,
          }),
        );
      }

      await client.send(new DeleteBucketCommand({ Bucket: bucket }));
      console.log(`Delete for ${bucket} complete.`);
    }
  });

export { confirmCleanup, cleanupAction };
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for JavaScript .
  - GetObjectLegalHold
  - GetObjectLockConfiguration
  - GetObjectRetention
  - PutObjectLegalHold
  - PutObjectLockConfiguration
  - PutObjectRetention

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Gérez les listes de contrôle d'accès (ACL) pour les compartiments Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment gérer les listes de contrôle d'accès (ACL) pour les compartiments Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to manage Amazon Simple Storage Service
/// (Amazon S3) access control lists (ACLs) to control Amazon S3 bucket
/// access.
/// </summary>
public class ManageACLs
{
    public static async Task Main()
    {
        string bucketName = "doc-example-bucket1";
        string newBucketName = "doc-example-bucket2";
        string keyName = "sample-object.txt";
        string emailAddress = "someone@example.com";

        // If the AWS Region where your bucket is located is different from
        // the Region defined for the default user, pass the Amazon S3
bucket's
        // name to the client constructor. It should look like this:
        // RegionEndpoint bucketRegion = RegionEndpoint.USEast1;
```

```
            IAmazonS3 client = new AmazonS3Client();

            await TestBucketObjectACLsAsync(client, bucketName, newBucketName,
keyName, emailAddress);
        }

        /// <summary>
        /// Creates a new Amazon S3 bucket with a canned ACL, then retrieves the
ACL
        /// information and then adds a new ACL to one of the objects in the
        /// Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
call
        /// methods to create a bucket, get an ACL, and add a different ACL to
        /// one of the objects.</param>
        /// <param name="bucketName">A string representing the original Amazon S3
        /// bucket name.</param>
        /// <param name="newBucketName">A string representing the name of the
        /// new bucket that will be created.</param>
        /// <param name="keyName">A string representing the key name of an Amazon
S3
        /// object for which we will change the ACL.</param>
        /// <param name="emailAddress">A string representing the email address
        /// belonging to the person to whom access to the Amazon S3 bucket will
be
        /// granted.</param>
        public static async Task TestBucketObjectACLsAsync(
            IAmazonS3 client,
            string bucketName,
            string newBucketName,
            string keyName,
            string emailAddress)
        {
            try
            {
                // Create a new Amazon S3 bucket and specify canned ACL.
                var success = await CreateBucketWithCannedACLAsync(client,
newBucketName);

                // Get the ACL on a bucket.
                await GetBucketACLAsync(client, bucketName);

                // Add (replace) the ACL on an object in a bucket.
```

```
                await AddACLToExistingObjectAsync(client, bucketName, keyName,
 emailAddress);
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine($"Exception: {amazonS3Exception.Message}");
            }
        }

        /// <summary>
        /// Creates a new Amazon S3 bucket with a canned ACL attached.
        /// </summary>
        /// <param name="client">The initialized client object used to call
        /// PutBucketAsync.</param>
        /// <param name="newBucketName">A string representing the name of the
        /// new Amazon S3 bucket.</param>
        /// <returns>Returns a boolean value indicating success or failure.</
returns>
        public static async Task<bool> CreateBucketWithCannedACLAsync(IAmazonS3
 client, string newBucketName)
        {
            var request = new PutBucketRequest()
            {
                BucketName = newBucketName,
                BucketRegion = S3Region.EUWest1,

                // Add a canned ACL.
                CannedACL = S3CannedACL.LogDeliveryWrite,
            };

            var response = await client.PutBucketAsync(request);
            return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
        }


        /// <summary>
        /// Retrieves the ACL associated with the Amazon S3 bucket name in the
        /// bucketName parameter.
        /// </summary>
        /// <param name="client">The initialized client object used to call
        /// PutBucketAsync.</param>
        /// <param name="bucketName">The Amazon S3 bucket for which we want to
 get the
        /// ACL list.</param>
```

```
        /// <returns>Returns an S3AccessControlList returned from the call to
        /// GetACLAsync.</returns>
        public static async Task<S3AccessControlList> GetBucketACLAsync(IAmazonS3
 client, string bucketName)
        {
            GetACLResponse response = await client.GetACLAsync(new GetACLRequest
            {
                BucketName = bucketName,
            });

            return response.AccessControlList;
        }



        /// <summary>
        /// Adds a new ACL to an existing object in the Amazon S3 bucket.
        /// </summary>
        /// <param name="client">The initialized client object used to call
        /// PutBucketAsync.</param>
        /// <param name="bucketName">A string representing the name of the Amazon
 S3
        /// bucket containing the object to which we want to apply a new ACL.</
param>
        /// <param name="keyName">A string representing the name of the object
        /// to which we want to apply the new ACL.</param>
        /// <param name="emailAddress">The email address of the person to whom
        /// we will be applying to whom access will be granted.</param>
        public static async Task AddACLToExistingObjectAsync(IAmazonS3 client,
 string bucketName, string keyName, string emailAddress)
        {
            // Retrieve the ACL for an object.
            GetACLResponse aclResponse = await client.GetACLAsync(new
 GetACLRequest
            {
                BucketName = bucketName,
                Key = keyName,
            });

            S3AccessControlList acl = aclResponse.AccessControlList;

            // Retrieve the owner.
            Owner owner = acl.Owner;
```

```
            // Clear existing grants.
            acl.Grants.Clear();

            // Add a grant to reset the owner's full permission
            // (the previous clear statement removed all permissions).
            var fullControlGrant = new S3Grant
            {
                Grantee = new S3Grantee { CanonicalUser = acl.Owner.Id },
            };
            acl.AddGrant(fullControlGrant.Grantee, S3Permission.FULL_CONTROL);

            // Specify email to identify grantee for granting permissions.
            var grantUsingEmail = new S3Grant
            {
                Grantee = new S3Grantee { EmailAddress = emailAddress },
                Permission = S3Permission.WRITE_ACP,
            };

            // Specify log delivery group as grantee.
            var grantLogDeliveryGroup = new S3Grant
            {
                Grantee = new S3Grantee { URI = "http://acs.amazonaws.com/groups/
s3/LogDelivery" },
                Permission = S3Permission.WRITE,
            };

            // Create a new ACL.
            var newAcl = new S3AccessControlList
            {
                Grants = new List<S3Grant> { grantUsingEmail,
 grantLogDeliveryGroup },
                Owner = owner,
            };

            // Set the new ACL. We're throwing away the response here.
            _ = await client.PutACLAsync(new PutACLRequest
            {
                BucketName = bucketName,
                Key = keyName,
                AccessControlList = newAcl,
            });
        }

    }
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - GetBucketAcl
  - GetObjectAcl
  - PutBucketAcl
  - PutObjectAcl

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Gérez des objets Amazon S3 versionnés par lots à l'aide d'une fonction Lambda à l'aide d'un SDK AWS

L'exemple de code suivant montre comment gérer des objets S3 soumis au contrôle de version par lots avec une fonction Lambda.

Python

SDK pour Python (Boto3)

Montre comment manipuler des objets versionnés par Amazon Simple Storage Service (Amazon S3) par lots en créant des tâches qui AWS Lambda appellent des fonctions pour effectuer le traitement. Cet exemple montre comment créer un compartiment compatible avec les versions et télécharge les strophes du poème Vous êtes vieux, Père Guillaume de Lewis Carroll. Il utilise également des tâches par lots Amazon S3 pour modifier le poème de différentes manières.

Découvrez comment :

- Créez des fonctions Lambda qui fonctionnent sur des objets soumis au contrôle de version.
- Créez un manifeste des objets à mettre à jour.
- Créez des tâches par lots qui appellent des fonctions Lambda pour mettre à jour les objets.
- Supprimez les fonctions Lambda.
- Videz et supprimez un compartiment soumis au contrôle de version.

Il est préférable de visionner cet exemple sur GitHub. Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon S3

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Analyser les URI Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment analyser des URI Amazon S3 pour extraire des composants importants tels que le nom du compartiment et la clé d'objet.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Analysez un URI Amazon S3 à l'aide de la classe S3Uri.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.S3Uri;
import software.amazon.awssdk.services.s3.S3Utilities;

import java.net.URI;
import java.util.List;
import java.util.Map;

    /**
     *
```

```
     * @param s3Client    - An S3Client through which you acquire an S3Uri
 instance.
     * @param s3ObjectUrl - A complex URL (String) that is used to demonstrate
 S3Uri
     *                    capabilities.
     */
    public static void parseS3UriExample(S3Client s3Client, String s3ObjectUrl) {
        logger.info(s3ObjectUrl);
        // Console output:
        // 'https://s3.us-west-1.amazonaws.com/myBucket/resources/doc.txt?
versionId=abc123&partNumber=77&partNumber=88'.

        // Create an S3Utilities object using the configuration of the s3Client.
        S3Utilities s3Utilities = s3Client.utilities();

        // From a String URL create a URI object to pass to the parseUri()
 method.
        URI uri = URI.create(s3ObjectUrl);
        S3Uri s3Uri = s3Utilities.parseUri(uri);

        // If the URI contains no value for the Region, bucket or key, the SDK
 returns
        // an empty Optional.
        // The SDK returns decoded URI values.

        Region region = s3Uri.region().orElse(null);
        log("region", region);
        // Console output: 'region: us-west-1'.

        String bucket = s3Uri.bucket().orElse(null);
        log("bucket", bucket);
        // Console output: 'bucket: myBucket'.

        String key = s3Uri.key().orElse(null);
        log("key", key);
        // Console output: 'key: resources/doc.txt'.

        Boolean isPathStyle = s3Uri.isPathStyle();
        log("isPathStyle", isPathStyle);
        // Console output: 'isPathStyle: true'.

        // If the URI contains no query parameters, the SDK returns an empty map.
        Map<String, List<String>> queryParams = s3Uri.rawQueryParameters();
        log("rawQueryParameters", queryParams);
```

```
        // Console output: 'rawQueryParameters: {versionId=[abc123],
 partNumber=[77,
        // 88]}'.

        // Retrieve the first or all values for a query parameter as shown in the
        // following code.
        String versionId =
 s3Uri.firstMatchingRawQueryParameter("versionId").orElse(null);
        log("firstMatchingRawQueryParameter-versionId", versionId);
        // Console output: 'firstMatchingRawQueryParameter-versionId: abc123'.

        String partNumber =
 s3Uri.firstMatchingRawQueryParameter("partNumber").orElse(null);
        log("firstMatchingRawQueryParameter-partNumber", partNumber);
        // Console output: 'firstMatchingRawQueryParameter-partNumber: 77'.

        List<String> partNumbers =
 s3Uri.firstMatchingRawQueryParameters("partNumber");
        log("firstMatchingRawQueryParameter", partNumbers);
        // Console output: 'firstMatchingRawQueryParameter: [77, 88]'.

        /*
         * Object keys and query parameters with reserved or unsafe characters,
 must be
         * URL-encoded.
         * For example replace whitespace " " with "%20".
         * Valid:
         * "https://s3.us-west-1.amazonaws.com/myBucket/object%20key?query=
%5Bbrackets%5D"
         * Invalid:
         * "https://s3.us-west-1.amazonaws.com/myBucket/object key?
query=[brackets]"
         *
         * Virtual-hosted-style URIs with bucket names that contain a dot, ".",
 the dot
         * must not be URL-encoded.
         * Valid: "https://my.Bucket.s3.us-west-1.amazonaws.com/key"
         * Invalid: "https://my%2EBucket.s3.us-west-1.amazonaws.com/key"
         */
    }

    private static void log(String s3UriElement, Object element) {
        if (element == null) {
            logger.info("{}: {}", s3UriElement, "null");
```

```
        } else {
            logger.info("{}: {}", s3UriElement, element);
        }
    }
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Réaliser une copie en plusieurs parties d'un objet Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment effectuer une copie en plusieurs parties d'un objet Amazon S3.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// This example shows how to perform a multi-part copy from one Amazon
/// Simple Storage Service (Amazon S3) bucket to another.
/// </summary>
public class MPUapiCopyObj
{
    private const string SourceBucket = "doc-example-bucket1";
    private const string TargetBucket = "doc-example-bucket2";
```

```
        private const string SourceObjectKey = "example.mov";
        private const string TargetObjectKey = "copied_video_file.mov";

        /// <summary>
        /// This method starts the multi-part upload.
        /// </summary>
        public static async Task Main()
        {
            var s3Client = new AmazonS3Client();
            Console.WriteLine("Copying object...");
            await MPUCopyObjectAsync(s3Client);
        }

        /// <summary>
        /// This method uses the passed client object to perform a multipart
        /// copy operation.
        /// </summary>
        /// <param name="client">An Amazon S3 client object that will be used
        /// to perform the copy.</param>
        public static async Task MPUCopyObjectAsync(AmazonS3Client client)
        {
            // Create a list to store the copy part responses.
            var copyResponses = new List<CopyPartResponse>();

            // Setup information required to initiate the multipart upload.
            var initiateRequest = new InitiateMultipartUploadRequest
            {
                BucketName = TargetBucket,
                Key = TargetObjectKey,
            };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await client.InitiateMultipartUploadAsync(initiateRequest);

            // Save the upload ID.
            string uploadId = initResponse.UploadId;

            try
            {
                // Get the size of the object.
                var metadataRequest = new GetObjectMetadataRequest
                {
                    BucketName = SourceBucket,
```

```
                    Key = SourceObjectKey,
                };

                GetObjectMetadataResponse metadataResponse =
                    await client.GetObjectMetadataAsync(metadataRequest);
                var objectSize = metadataResponse.ContentLength; // Length in
bytes.

                // Copy the parts.
                var partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

                long bytePosition = 0;
                for (int i = 1; bytePosition < objectSize; i++)
                {
                    var copyRequest = new CopyPartRequest
                    {
                        DestinationBucket = TargetBucket,
                        DestinationKey = TargetObjectKey,
                        SourceBucket = SourceBucket,
                        SourceKey = SourceObjectKey,
                        UploadId = uploadId,
                        FirstByte = bytePosition,
                        LastByte = bytePosition + partSize - 1 >= objectSize ?
objectSize - 1 : bytePosition + partSize - 1,
                        PartNumber = i,
                    };

                    copyResponses.Add(await client.CopyPartAsync(copyRequest));

                    bytePosition += partSize;
                }

                // Set up to complete the copy.
                var completeRequest = new CompleteMultipartUploadRequest
                {
                    BucketName = TargetBucket,
                    Key = TargetObjectKey,
                    UploadId = initResponse.UploadId,
                };
                completeRequest.AddPartETags(copyResponses);

                // Complete the copy.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await client.CompleteMultipartUploadAsync(completeRequest);
```

```
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error encountered on server.
Message:'{e.Message}' when writing an object");
            }
            catch (Exception e)
            {
                Console.WriteLine($"Unknown encountered on server.
Message:'{e.Message}' when writing an object");
            }
        }
    }
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - GetObjectMetadata
  - UploadPartCopy

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Effectuer un téléchargement partitionné d'un objet Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment effectuer un chargement partitionné vers un objet Amazon S3.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Les exemples de code utilisent les importations suivantes.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;
```

Utilisez le gestionnaire de transferts S3 situé au-dessus du client S3 basé sur AWS CRT pour effectuer de manière transparente un téléchargement partitionné lorsque la taille du contenu dépasse un seuil. La taille par défaut est de 8 Mo.

```
    public void multipartUploadWithTransferManager(String filePath) {
        S3TransferManager transferManager = S3TransferManager.create();
        UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
                .putObjectRequest(b -> b
                        .bucket(bucketName)
                        .key(key))
                .source(Paths.get(filePath))
                .build();
        FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
        fileUpload.completionFuture().join();
        transferManager.close();
    }
```

Utilisez l'API S3Client pour effectuer un téléchargement en plusieurs parties.

```
    public void multipartUploadWithS3Client(String filePath) {

        // Initiate the multipart upload.
        CreateMultipartUploadResponse createMultipartUploadResponse =
s3Client.createMultipartUpload(b -> b
                .bucket(bucketName)
                .key(key));
        String uploadId = createMultipartUploadResponse.uploadId();

        // Upload the parts of the file.
        int partNumber = 1;
        List<CompletedPart> completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            long position = 0;
            while (position < fileSize) {
                file.seek(position);
                long read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
```

```
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .partNumber(partNumber)
                        .build();

                UploadPartResponse partResponse = s3Client.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
                position += read;
                partNumber++;
            }
        } catch (IOException e) {
            logger.error(e.getMessage());
        }

        // Complete the multipart upload.
        s3Client.completeMultipartUpload(b -> b
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)

.multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));
    }
```

Utilisez l'AsyncClient API S3 avec le support multipartie activé pour effectuer un téléchargement partitionné.

```
    public void multipartUploadWithS3AsyncClient(String filePath) {
        // Enable multipart support.
        S3AsyncClient s3AsyncClient = S3AsyncClient.builder()
                .multipartEnabled(true)
```

```
                .build();

        CompletableFuture<PutObjectResponse> response = s3AsyncClient.putObject(b
-> b
                        .bucket(bucketName)
                        .key(key),
                Paths.get(filePath));

        response.join();
        logger.info("File uploaded in multiple 8 MiB parts using
S3AsyncClient.");
    }
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Suivez le chargement ou le téléchargement d'un objet Amazon S3 à l'aide d'un AWS SDK

L'exemple de code suivant montre comment suivre le chargement ou le téléchargement d'un objet Amazon S3.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Suivez la progression du téléchargement d'un fichier.

```
   public void trackUploadFile(S3TransferManager transferManager, String
bucketName,
                               String key, URI filePathURI) {
       UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
               .putObjectRequest(b -> b.bucket(bucketName).key(key))
               .addTransferListener(LoggingTransferListener.create())  // Add
listener.
               .source(Paths.get(filePathURI))
               .build();

       FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);

       fileUpload.completionFuture().join();
       /*
           The SDK provides a LoggingTransferListener implementation of the
TransferListener interface.
           You can also implement the interface to provide your own logic.

           Configure log4J2 with settings such as the following.
               <Configuration status="WARN">
                   <Appenders>
                       <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
                           <PatternLayout pattern="%m%n"/>
                       </Console>
                   </Appenders>

                   <Loggers>
                       <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
                           <AppenderRef ref="AlignedConsoleAppender"/>
                       </logger>
                   </Loggers>
               </Configuration>

           Log4J2 logs the progress. The following is example output for a 21.3
MB file upload.
               Transfer initiated...
               |                    | 0.0%
               |====                | 21.1%
               |============        | 60.5%
```

```
                |====================| 100.0%
                Transfer complete!
        */
    }
```

Suivez la progression du téléchargement d'un fichier.

```
   public void trackDownloadFile(S3TransferManager transferManager, String
bucketName,
                              String key, String downloadedFileWithPath) {
        DownloadFileRequest downloadFileRequest = DownloadFileRequest.builder()
                .getObjectRequest(b -> b.bucket(bucketName).key(key))
                .addTransferListener(LoggingTransferListener.create())  // Add
listener.
                .destination(Paths.get(downloadedFileWithPath))
                .build();

        FileDownload downloadFile =
transferManager.downloadFile(downloadFileRequest);

        CompletedFileDownload downloadResult =
downloadFile.completionFuture().join();
        /*
            The SDK provides a LoggingTransferListener implementation of the
TransferListener interface.
            You can also implement the interface to provide your own logic.

            Configure log4J2 with settings such as the following.
                <Configuration status="WARN">
                    <Appenders>
                        <Console name="AlignedConsoleAppender"
target="SYSTEM_OUT">
                            <PatternLayout pattern="%m%n"/>
                        </Console>
                    </Appenders>

                    <Loggers>
                        <logger
name="software.amazon.awssdk.transfer.s3.progress.LoggingTransferListener"
level="INFO" additivity="false">
                            <AppenderRef ref="AlignedConsoleAppender"/>
                        </logger>
```

```
                </Loggers>
            </Configuration>

        Log4J2 logs the progress. The following is example output for a 21.3
MB file download.
            Transfer initiated...
            |=======             | 39.4%
            |===============     | 78.8%
            |====================| 100.0%
            Transfer complete!
        */
    }
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - GetObject
  - PutObject

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples d'approches pour les tests unitaires et d'intégration avec un AWS SDK

L'exemple de code suivant montre comment utiliser des exemples de meilleures pratiques lors de l'écriture de tests unitaires et d'intégration à l'aide d'un AWS SDK.

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Cargo.toml pour tester des exemples.

```
[package]
name = "testing-examples"
version = "0.1.0"
authors = [
  "John Disanti <jdisanti@amazon.com>",
  "Doug Schwartz <dougsch@amazon.com>",
]
edition = "2021"

# snippet-start:[testing.rust.Cargo.toml]
[dependencies]
async-trait = "0.1.51"
aws-config = { version = "1.0.1", features = ["behavior-version-latest"] }
aws-credential-types = { version = "1.0.1", features = [ "hardcoded-
credentials", ] }
aws-sdk-s3 = { version = "1.4.0" }
aws-smithy-types = { version = "1.0.1" }
aws-smithy-runtime = { version = "1.0.1", features = ["test-util"] }
aws-smithy-runtime-api = { version = "1.0.1", features = ["test-util"] }
aws-types = { version = "1.0.1" }
clap = { version = "~4.4", features = ["derive"] }
http = "0.2.9"
mockall = "0.11.4"
serde_json = "1"
tokio = { version = "1.20.1", features = ["full"] }
tracing-subscriber = { version = "0.3.15", features = ["env-filter"] }
# snippet-end:[testing.rust.Cargo.toml]

[[bin]]
name = "main"
path = "src/main.rs"
```

Exemple de test unitaire à l'aide d'une simulation automatique et d'un encapsuleur de service.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

// snippet-start:[testing.rust.wrapper]
// snippet-start:[testing.rust.wrapper-uses]
use aws_sdk_s3 as s3;
#[allow(unused_imports)]
use mockall::automock;
```

```
use s3::operation::list_objects_v2::{ListObjectsV2Error, ListObjectsV2Output};
// snippet-end:[testing.rust.wrapper-uses]

// snippet-start:[testing.rust.wrapper-which-impl]
#[cfg(test)]
pub use MockS3Impl as S3;
#[cfg(not(test))]
pub use S3Impl as S3;
// snippet-end:[testing.rust.wrapper-which-impl]

// snippet-start:[testing.rust.wrapper-impl]
#[allow(dead_code)]
pub struct S3Impl {
    inner: s3::Client,
}

#[cfg_attr(test, automock)]
impl S3Impl {
    #[allow(dead_code)]
    pub fn new(inner: s3::Client) -> Self {
        Self { inner }
    }

    #[allow(dead_code)]
    pub async fn list_objects(
        &self,
        bucket: &str,
        prefix: &str,
        continuation_token: Option<String>,
    ) -> Result<ListObjectsV2Output, s3::error::SdkError<ListObjectsV2Error>> {
        self.inner
            .list_objects_v2()
            .bucket(bucket)
            .prefix(prefix)
            .set_continuation_token(continuation_token)
            .send()
            .await
    }
}
// snippet-end:[testing.rust.wrapper-impl]

// snippet-start:[testing.rust.wrapper-func]
#[allow(dead_code)]
```

```
pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3_list: S3,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3_list
            .list_objects(bucket, prefix, next_token.take())
            .await?;

        // Add up the file sizes we got back
        for object in result.contents() {
            total_size_bytes += object.size().unwrap_or(0) as usize;
        }

        // Handle pagination, and break the loop if there are no more pages
        next_token = result.next_continuation_token.clone();
        if next_token.is_none() {
            break;
        }
    }
    Ok(total_size_bytes)
}
// snippet-end:[testing.rust.wrapper-func]
// snippet-end:[testing.rust.wrapper]

// snippet-start:[testing.rust.wrapper-test-mod]
#[cfg(test)]
mod test {
    // snippet-start:[testing.rust.wrapper-tests]
    use super::*;
    use mockall::predicate::eq;

    // snippet-start:[testing.rust.wrapper-test-single]
    #[tokio::test]
    async fn test_single_page() {
        let mut mock = MockS3Impl::default();
        mock.expect_list_objects()
            .with(eq("test-bucket"), eq("test-prefix"), eq(None))
            .return_once(|_, _, _| {
```

```
                Ok(ListObjectsV2Output::builder()
                    .set_contents(Some(vec![
                        // Mock content for ListObjectsV2 response
                        s3::types::Object::builder().size(5).build(),
                        s3::types::Object::builder().size(2).build(),
                    ]))
                    .build())
            });

        // Run the code we want to test with it
        let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
            .await
            .unwrap();

        // Verify we got the correct total size back
        assert_eq!(7, size);
    }
    // snippet-end:[testing.rust.wrapper-test-single]

    // snippet-start:[testing.rust.wrapper-test-multiple]
    #[tokio::test]
    async fn test_multiple_pages() {
        // Create the Mock instance with two pages of objects now
        let mut mock = MockS3Impl::default();
        mock.expect_list_objects()
            .with(eq("test-bucket"), eq("test-prefix"), eq(None))
            .return_once(|_, _, _| {
                Ok(ListObjectsV2Output::builder()
                    .set_contents(Some(vec![
                        // Mock content for ListObjectsV2 response
                        s3::types::Object::builder().size(5).build(),
                        s3::types::Object::builder().size(2).build(),
                    ]))
                    .set_next_continuation_token(Some("next".to_string()))
                    .build())
            });
        mock.expect_list_objects()
            .with(
                eq("test-bucket"),
                eq("test-prefix"),
                eq(Some("next".to_string())),
            )
            .return_once(|_, _, _| {
                Ok(ListObjectsV2Output::builder()
```

```
                    .set_contents(Some(vec![
                        // Mock content for ListObjectsV2 response
                        s3::types::Object::builder().size(3).build(),
                        s3::types::Object::builder().size(9).build(),
                    ]))
                    .build())
            });

        // Run the code we want to test with it
        let size = determine_prefix_file_size(mock, "test-bucket", "test-prefix")
            .await
            .unwrap();

        assert_eq!(19, size);
    }
    // snippet-end:[testing.rust.wrapper-test-multiple]
    // snippet-end:[testing.rust.wrapper-tests]
}
// snippet-end:[testing.rust.wrapper-test-mod]
```

Exemple de test d'intégration utilisant StaticReplayClient.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

// snippet-start:[testing.rust.replay-uses]
use aws_sdk_s3 as s3;
// snippet-end:[testing.rust.replay-uses]

#[allow(dead_code)]
// snippet-start:[testing.rust.replay]
pub async fn determine_prefix_file_size(
    // Now we take a reference to our trait object instead of the S3 client
    // s3_list: ListObjectsService,
    s3: s3::Client,
    bucket: &str,
    prefix: &str,
) -> Result<usize, s3::Error> {
    let mut next_token: Option<String> = None;
    let mut total_size_bytes = 0;
    loop {
        let result = s3
```

```
            .list_objects_v2()
            .prefix(prefix)
            .bucket(bucket)
            .set_continuation_token(next_token.take())
            .send()
            .await?;

        // Add up the file sizes we got back
        for object in result.contents() {
            total_size_bytes += object.size().unwrap_or(0) as usize;
        }

        // Handle pagination, and break the loop if there are no more pages
        next_token = result.next_continuation_token.clone();
        if next_token.is_none() {
            break;
        }
    }
    Ok(total_size_bytes)
}
// snippet-end:[testing.rust.replay]

#[allow(dead_code)]
// snippet-start:[testing.rust.replay-tests]
// snippet-start:[testing.rust.replay-make-credentials]
fn make_s3_test_credentials() -> s3::config::Credentials {
    s3::config::Credentials::new(
        "ATESTCLIENT",
        "astestsecretkey",
        Some("atestsessiontoken".to_string()),
        None,
        "",
    )
}
// snippet-end:[testing.rust.replay-make-credentials]

// snippet-start:[testing.rust.replay-test-module]
#[cfg(test)]
mod test {
    // snippet-start:[testing.rust.replay-test-single]
    use super::*;
    use aws_config::BehaviorVersion;
    use aws_sdk_s3 as s3;
```

```
    use aws_smithy_runtime::client::http::test_util::{ReplayEvent,
 StaticReplayClient};
    use aws_smithy_types::body::SdkBody;

    #[tokio::test]
    async fn test_single_page() {
        let page_1 = ReplayEvent::new(
                http::Request::builder()
                    .method("GET")
                    .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
                    .body(SdkBody::empty())
                    .unwrap(),
                http::Response::builder()
                    .status(200)
                    .body(SdkBody::from(include_str!("./testing/
response_1.xml")))
                    .unwrap(),
            );
        let replay_client = StaticReplayClient::new(vec![page_1]);
        let client: s3::Client = s3::Client::from_conf(
            s3::Config::builder()
                .behavior_version(BehaviorVersion::latest())
                .credentials_provider(make_s3_test_credentials())
                .region(s3::config::Region::new("us-east-1"))
                .http_client(replay_client.clone())
                .build(),
        );

        // Run the code we want to test with it
        let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")
            .await
            .unwrap();

        // Verify we got the correct total size back
        assert_eq!(7, size);
        replay_client.assert_requests_match(&[]);
    }
    // snippet-end:[testing.rust.replay-test-single]

    // snippet-start:[testing.rust.replay-test-multiple]
    #[tokio::test]
    async fn test_multiple_pages() {
```

```
        // snippet-start:[testing.rust.replay-create-replay]
        let page_1 = ReplayEvent::new(
                http::Request::builder()
                    .method("GET")
                    .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix")
                    .body(SdkBody::empty())
                    .unwrap(),
                http::Response::builder()
                    .status(200)
                    .body(SdkBody::from(include_str!("./testing/
response_multi_1.xml")))
                    .unwrap(),
            );
        let page_2 = ReplayEvent::new(
                http::Request::builder()
                    .method("GET")
                    .uri("https://test-bucket.s3.us-east-1.amazonaws.com/?list-
type=2&prefix=test-prefix&continuation-token=next")
                    .body(SdkBody::empty())
                    .unwrap(),
                http::Response::builder()
                    .status(200)
                    .body(SdkBody::from(include_str!("./testing/
response_multi_2.xml")))
                    .unwrap(),
            );
        let replay_client = StaticReplayClient::new(vec![page_1, page_2]);
        // snippet-end:[testing.rust.replay-create-replay]
        // snippet-start:[testing.rust.replay-create-client]
        let client: s3::Client = s3::Client::from_conf(
            s3::Config::builder()
                .behavior_version(BehaviorVersion::latest())
                .credentials_provider(make_s3_test_credentials())
                .region(s3::config::Region::new("us-east-1"))
                .http_client(replay_client.clone())
                .build(),
        );
        // snippet-end:[testing.rust.replay-create-client]

        // Run the code we want to test with it
        // snippet-start:[testing.rust.replay-test-and-verify]
        let size = determine_prefix_file_size(client, "test-bucket", "test-
prefix")
```

```
            .await
            .unwrap();

        assert_eq!(19, size);

        replay_client.assert_requests_match(&[]);
        // snippet-end:[testing.rust.replay-test-and-verify]
    }
    // snippet-end:[testing.rust.replay-test-multiple]
}
// snippet-end:[testing.rust.replay-tests]
// snippet-end:[testing.rust.replay-test-module]
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Charger récursivement un répertoire local dans un compartiment Amazon Simple Storage Service (Amazon S3)

L'exemple de code suivant montre comment charger un répertoire local de manière récursive dans un compartiment Amazon Simple Storage Service (Amazon S3).

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Utilisez un S3 TransferManager pour télécharger un répertoire local. Consultez le fichier complet et le test.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.s3.model.ObjectIdentifier;
```

```
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedDirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.DirectoryUpload;
import software.amazon.awssdk.transfer.s3.model.UploadDirectoryRequest;

import java.net.URI;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.file.Paths;
import java.util.UUID;

    public Integer uploadDirectory(S3TransferManager transferManager,
            URI sourceDirectory, String bucketName) {
        DirectoryUpload directoryUpload =
 transferManager.uploadDirectory(UploadDirectoryRequest.builder()
                .source(Paths.get(sourceDirectory))
                .bucket(bucketName)
                .build());

        CompletedDirectoryUpload completedDirectoryUpload =
 directoryUpload.completionFuture().join();
        completedDirectoryUpload.failedTransfers()
                .forEach(fail -> logger.warn("Object [{}] failed to transfer",
 fail.toString()));
        return completedDirectoryUpload.failedTransfers().size();
    }
```

- Pour plus de détails sur l'API, reportez-vous UploadDirectoryà la section Référence des AWS SDK for Java 2.x API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Chargez ou téléchargez des fichiers volumineux vers et depuis Amazon S3 à l'aide d'un AWS SDK

Les exemples de code suivants montrent comment charger ou télécharger des fichiers volumineux vers et depuis Amazon S3.

Pour plus d'informations, consultez la rubrique Uploading an object using multipart upload (Chargement d'un objet à l'aide du chargement partitionné).

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Appelez des fonctions qui transfèrent des fichiers vers et depuis un compartiment S3 à l'aide d'Amazon S3 TransferUtility.

```
global using System.Text;
global using Amazon.S3;
global using Amazon.S3.Model;
global using Amazon.S3.Transfer;
global using TransferUtilityBasics;



// This Amazon S3 client uses the default user credentials
// defined for this computer.
using Microsoft.Extensions.Configuration;

IAmazonS3 client = new AmazonS3Client();
var transferUtil = new TransferUtility(client);
IConfiguration _configuration;

_configuration = new ConfigurationBuilder()
    .SetBasePath(Directory.GetCurrentDirectory())
    .AddJsonFile("settings.json") // Load test settings from JSON file.
    .AddJsonFile("settings.local.json",
        true) // Optionally load local settings.
    .Build();

// Edit the values in settings.json to use an S3 bucket and files that
// exist on your AWS account and on the local computer where you
// run this scenario.
```

```
var bucketName = _configuration["BucketName"];
var localPath =
 $"{Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)}\
\TransferFolder";

DisplayInstructions();

PressEnter();

Console.WriteLine();

// Upload a single file to an S3 bucket.
DisplayTitle("Upload a single file");

var fileToUpload = _configuration["FileToUpload"];
Console.WriteLine($"Uploading {fileToUpload} to the S3 bucket, {bucketName}.");

var success = await TransferMethods.UploadSingleFileAsync(transferUtil,
 bucketName, fileToUpload, localPath);
if (success)
{
    Console.WriteLine($"Successfully uploaded the file, {fileToUpload} to
 {bucketName}.");
}

PressEnter();

// Upload a local directory to an S3 bucket.
DisplayTitle("Upload all files from a local directory");
Console.WriteLine("Upload all the files in a local folder to an S3 bucket.");
const string keyPrefix = "UploadFolder";
var uploadPath = $"{localPath}\\UploadFolder";

Console.WriteLine($"Uploading the files in {uploadPath} to {bucketName}");
DisplayTitle($"{uploadPath} files");
DisplayLocalFiles(uploadPath);
Console.WriteLine();

PressEnter();

success = await TransferMethods.UploadFullDirectoryAsync(transferUtil,
 bucketName, keyPrefix, uploadPath);
if (success)
{
```

```
    Console.WriteLine($"Successfully uploaded the files in {uploadPath} to
 {bucketName}.");
    Console.WriteLine($"{bucketName} currently contains the following files:");
    await DisplayBucketFiles(client, bucketName, keyPrefix);
    Console.WriteLine();
}

PressEnter();

// Download a single file from an S3 bucket.
DisplayTitle("Download a single file");
Console.WriteLine("Now we will download a single file from an S3 bucket.");

var keyName = _configuration["FileToDownload"];

Console.WriteLine($"Downloading {keyName} from {bucketName}.");

success = await TransferMethods.DownloadSingleFileAsync(transferUtil, bucketName,
 keyName, localPath);
if (success)
{
    Console.WriteLine("$Successfully downloaded the file, {keyName} from
 {bucketName}.");
}

PressEnter();

// Download the contents of a directory from an S3 bucket.
DisplayTitle("Download the contents of an S3 bucket");
var s3Path = _configuration["S3Path"];
var downloadPath = $"{localPath}\\{s3Path}";

Console.WriteLine($"Downloading the contents of {bucketName}\\{s3Path}");
Console.WriteLine($"{bucketName}\\{s3Path} contains the following files:");
await DisplayBucketFiles(client, bucketName, s3Path);
Console.WriteLine();

success = await TransferMethods.DownloadS3DirectoryAsync(transferUtil,
 bucketName, s3Path, downloadPath);
if (success)
{
    Console.WriteLine($"Downloaded the files in {bucketName} to
 {downloadPath}.");
    Console.WriteLine($"{downloadPath} now contains the following files:");
```

```
    DisplayLocalFiles(downloadPath);
}

Console.WriteLine("\nThe TransferUtility Basics application has completed.");
PressEnter();

// Displays the title for a section of the scenario.
static void DisplayTitle(string titleText)
{
    var sepBar = new string('-', Console.WindowWidth);

    Console.WriteLine(sepBar);
    Console.WriteLine(CenterText(titleText));
    Console.WriteLine(sepBar);
}

// Displays a description of the actions to be performed by the scenario.
static void DisplayInstructions()
{
    var sepBar = new string('-', Console.WindowWidth);

    DisplayTitle("Amazon S3 Transfer Utility Basics");
    Console.WriteLine("This program shows how to use the Amazon S3 Transfer
 Utility.");
    Console.WriteLine("It performs the following actions:");
    Console.WriteLine("\t1. Upload a single object to an S3 bucket.");
    Console.WriteLine("\t2. Upload an entire directory from the local computer to
 an\n\t  S3 bucket.");
    Console.WriteLine("\t3. Download a single object from an S3 bucket.");
    Console.WriteLine("\t4. Download the objects in an S3 bucket to a local
 directory.");
    Console.WriteLine($"\n{sepBar}");
}

// Pauses the scenario.
static void PressEnter()
{
    Console.WriteLine("Press <Enter> to continue.");
    _ = Console.ReadLine();
    Console.WriteLine("\n");
}

// Returns the string textToCenter, padded on the left with spaces
// that center the text on the console display.
```

```
static string CenterText(string textToCenter)
{
    var centeredText = new StringBuilder();
    var screenWidth = Console.WindowWidth;
    centeredText.Append(new string(' ', (int)(screenWidth -
 textToCenter.Length) / 2));
    centeredText.Append(textToCenter);
    return centeredText.ToString();
}

// Displays a list of file names included in the specified path.
static void DisplayLocalFiles(string localPath)
{
    var fileList = Directory.GetFiles(localPath);
    if (fileList.Length > 0)
    {
        foreach (var fileName in fileList)
        {
            Console.WriteLine(fileName);
        }
    }
}

// Displays a list of the files in the specified S3 bucket and prefix.
static async Task DisplayBucketFiles(IAmazonS3 client, string bucketName, string
 s3Path)
{
    ListObjectsV2Request request = new()
    {
        BucketName = bucketName,
        Prefix = s3Path,
        MaxKeys = 5,
    };

    var response = new ListObjectsV2Response();

    do
    {
        response = await client.ListObjectsV2Async(request);

        response.S3Objects
            .ForEach(obj => Console.WriteLine($"{obj.Key}"));

        // If the response is truncated, set the request ContinuationToken
```

```
        // from the NextContinuationToken property of the response.
        request.ContinuationToken = response.NextContinuationToken;
    } while (response.IsTruncated);
}
```

Chargez un seul fichier.

```
        /// <summary>
        /// Uploads a single file from the local computer to an S3 bucket.
        /// </summary>
        /// <param name="transferUtil">The transfer initialized TransferUtility
        /// object.</param>
        /// <param name="bucketName">The name of the S3 bucket where the file
        /// will be stored.</param>
        /// <param name="fileName">The name of the file to upload.</param>
        /// <param name="localPath">The local path where the file is stored.</
param>
        /// <returns>A boolean value indicating the success of the action.</
returns>
        public static async Task<bool> UploadSingleFileAsync(
            TransferUtility transferUtil,
            string bucketName,
            string fileName,
            string localPath)
        {
            if (File.Exists($"{localPath}\\{fileName}"))
            {
                try
                {
                    await transferUtil.UploadAsync(new
 TransferUtilityUploadRequest
                    {
                        BucketName = bucketName,
                        Key = fileName,
                        FilePath = $"{localPath}\\{fileName}",
                    });

                    return true;
                }
                catch (AmazonS3Exception s3Ex)
```

```
                {
                    Console.WriteLine($"Could not upload {fileName} from
 {localPath} because:");
                    Console.WriteLine(s3Ex.Message);
                    return false;
                }
            }
            else
            {
                Console.WriteLine($"{fileName} does not exist in {localPath}");
                return false;
            }
        }
```

Chargez un répertoire local complet.

```
        /// <summary>
        /// Uploads all the files in a local directory to a directory in an S3
        /// bucket.
        /// </summary>
        /// <param name="transferUtil">The transfer initialized TransferUtility
        /// object.</param>
        /// <param name="bucketName">The name of the S3 bucket where the files
        /// will be stored.</param>
        /// <param name="keyPrefix">The key prefix is the S3 directory where
        /// the files will be stored.</param>
        /// <param name="localPath">The local directory that contains the files
        /// to be uploaded.</param>
        /// <returns>A Boolean value representing the success of the action.</
returns>
        public static async Task<bool> UploadFullDirectoryAsync(
            TransferUtility transferUtil,
            string bucketName,
            string keyPrefix,
            string localPath)
        {
            if (Directory.Exists(localPath))
            {
                try
                {
```

```
                    await transferUtil.UploadDirectoryAsync(new
 TransferUtilityUploadDirectoryRequest
                    {
                        BucketName = bucketName,
                        KeyPrefix = keyPrefix,
                        Directory = localPath,
                    });

                    return true;
                }
                catch (AmazonS3Exception s3Ex)
                {
                    Console.WriteLine($"Can't upload the contents of {localPath}
 because:");
                    Console.WriteLine(s3Ex?.Message);
                    return false;
                }
            }
            else
            {
                Console.WriteLine($"The directory {localPath} does not exist.");
                return false;
            }
        }
```

Téléchargez un seul fichier.

```
        /// <summary>
        /// Download a single file from an S3 bucket to the local computer.
        /// </summary>
        /// <param name="transferUtil">The transfer initialized TransferUtility
        /// object.</param>
        /// <param name="bucketName">The name of the S3 bucket containing the
        /// file to download.</param>
        /// <param name="keyName">The name of the file to download.</param>
        /// <param name="localPath">The path on the local computer where the
        /// downloaded file will be saved.</param>
        /// <returns>A Boolean value indicating the results of the action.</
returns>
        public static async Task<bool> DownloadSingleFileAsync(
```

```
        TransferUtility transferUtil,
            string bucketName,
            string keyName,
            string localPath)
        {
            await transferUtil.DownloadAsync(new TransferUtilityDownloadRequest
            {
                BucketName = bucketName,
                Key = keyName,
                FilePath = $"{localPath}\\{keyName}",
            });

            return (File.Exists($"{localPath}\\{keyName}"));
        }
```

Téléchargez le contenu d'un compartiment S3.

```
        /// <summary>
        /// Downloads the contents of a directory in an S3 bucket to a
        /// directory on the local computer.
        /// </summary>
        /// <param name="transferUtil">The transfer initialized TransferUtility
        /// object.</param>
        /// <param name="bucketName">The bucket containing the files to
 download.</param>
        /// <param name="s3Path">The S3 directory where the files are located.</
param>
        /// <param name="localPath">The local path to which the files will be
        /// saved.</param>
        /// <returns>A Boolean value representing the success of the action.</
returns>
        public static async Task<bool> DownloadS3DirectoryAsync(
            TransferUtility transferUtil,
            string bucketName,
            string s3Path,
            string localPath)
        {
            int fileCount = 0;

            // If the directory doesn't exist, it will be created.
```

```
            if (Directory.Exists(s3Path))
            {
                var files = Directory.GetFiles(localPath);
                fileCount = files.Length;
            }

            await transferUtil.DownloadDirectoryAsync(new
TransferUtilityDownloadDirectoryRequest
            {
                BucketName = bucketName,
                LocalDirectory = localPath,
                S3Directory = s3Path,
            });

            if (Directory.Exists(localPath))
            {
                var files = Directory.GetFiles(localPath);
                if (files.Length > fileCount)
                {
                    return true;
                }

                // No change in the number of files. Assume
                // the download failed.
                return false;
            }

            // The local directory doesn't exist. No files
            // were downloaded.
            return false;
        }
```

Suivez la progression d'un téléchargement à l'aide du TransferUtility.

```
    using System;
    using System.Threading.Tasks;
    using Amazon.S3;
    using Amazon.S3.Transfer;

    /// <summary>
    /// This example shows how to track the progress of a multipart upload
```

```
    /// using the Amazon Simple Storage Service (Amazon S3) TransferUtility to
    /// upload to an Amazon S3 bucket.
    /// </summary>
    public class TrackMPUUsingHighLevelAPI
    {
        public static async Task Main()
        {
            string bucketName = "doc-example-bucket";
            string keyName = "sample_pic.png";
            string path = "filepath/directory/";
            string filePath = $"{path}{keyName}";

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
            IAmazonS3 client = new AmazonS3Client();

            await TrackMPUAsync(client, bucketName, filePath, keyName);
        }

        /// <summary>
        /// Starts an Amazon S3 multipart upload and assigns an event handler to
        /// track the progress of the upload.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client object used to
        /// perform the multipart upload.</param>
        /// <param name="bucketName">The name of the bucket to which to upload
        /// the file.</param>
        /// <param name="filePath">The path, including the file name of the
        /// file to be uploaded to the Amazon S3 bucket.</param>
        /// <param name="keyName">The file name to be used in the
        /// destination Amazon S3 bucket.</param>
        public static async Task TrackMPUAsync(
            IAmazonS3 client,
            string bucketName,
            string filePath,
            string keyName)
        {
            try
            {
                var fileTransferUtility = new TransferUtility(client);

                // Use TransferUtilityUploadRequest to configure options.
```

```
                // In this example we subscribe to an event.
                var uploadRequest =
                    new TransferUtilityUploadRequest
                    {
                        BucketName = bucketName,
                        FilePath = filePath,
                        Key = keyName,
                    };

                uploadRequest.UploadProgressEvent +=
                    new EventHandler<UploadProgressArgs>(
                        UploadRequest_UploadPartProgressEvent);

                await fileTransferUtility.UploadAsync(uploadRequest);
                Console.WriteLine("Upload completed");
            }
            catch (AmazonS3Exception ex)
            {
                Console.WriteLine($"Error:: {ex.Message}");
            }
        }

        /// <summary>
        /// Event handler to check the progress of the multipart upload.
        /// </summary>
        /// <param name="sender">The object that raised the event.</param>
        /// <param name="e">The object that contains multipart upload
        /// information.</param>
        public static void UploadRequest_UploadPartProgressEvent(object sender,
UploadProgressArgs e)
        {
            // Process event.
            Console.WriteLine($"{e.TransferredBytes}/{e.TotalBytes}");
        }
    }
```

Chargez un objet avec chiffrement.

```
    using System;
    using System.Collections.Generic;
    using System.IO;
```

```
using System.Security.Cryptography;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;

/// <summary>
/// Uses the Amazon Simple Storage Service (Amazon S3) low level API to
/// perform a multipart upload to an Amazon S3 bucket.
/// </summary>
public class SSECLowLevelMPUcopyObject
{
    public static async Task Main()
    {
        string existingBucketName = "doc-example-bucket";
        string sourceKeyName = "sample_file.txt";
        string targetKeyName = "sample_file_copy.txt";
        string filePath = $"sample\\{targetKeyName}";

        // If the AWS Region defined for your default user is different
        // from the Region where your Amazon S3 bucket is located,
        // pass the Region name to the Amazon S3 client object's constructor.
        // For example: RegionEndpoint.USEast1.
        IAmazonS3 client = new AmazonS3Client();

        // Create the encryption key.
        var base64Key = CreateEncryptionKey();

        await CreateSampleObjUsingClientEncryptionKeyAsync(
            client,
            existingBucketName,
            sourceKeyName,
            filePath,
            base64Key);
    }

    /// <summary>
    /// Creates the encryption key to use with the multipart upload.
    /// </summary>
    /// <returns>A string containing the base64-encoded key for encrypting
    /// the multipart upload.</returns>
    public static string CreateEncryptionKey()
    {
        Aes aesEncryption = Aes.Create();
        aesEncryption.KeySize = 256;
```

```
            aesEncryption.GenerateKey();
            string base64Key = Convert.ToBase64String(aesEncryption.Key);
            return base64Key;
        }

        /// <summary>
        /// Creates and uploads an object using a multipart upload.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 object used to
        /// initialize and perform the multipart upload.</param>
        /// <param name="existingBucketName">The name of the bucket to which
        /// the object will be uploaded.</param>
        /// <param name="sourceKeyName">The source object name.</param>
        /// <param name="filePath">The location of the source object.</param>
        /// <param name="base64Key">The encryption key to use with the upload.</
param>
        public static async Task CreateSampleObjUsingClientEncryptionKeyAsync(
            IAmazonS3 client,
            string existingBucketName,
            string sourceKeyName,
            string filePath,
            string base64Key)
        {
            List<UploadPartResponse> uploadResponses = new
 List<UploadPartResponse>();

            InitiateMultipartUploadRequest initiateRequest = new
 InitiateMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            InitiateMultipartUploadResponse initResponse =
               await client.InitiateMultipartUploadAsync(initiateRequest);

            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
```

```
                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                    {
                        BucketName = existingBucketName,
                        Key = sourceKeyName,
                        UploadId = initResponse.UploadId,
                        PartNumber = i,
                        PartSize = partSize,
                        FilePosition = filePosition,
                        FilePath = filePath,
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key,
                    };

                    // Upload part and add response to our list.
                    uploadResponses.Add(await
client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }

                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    UploadId = initResponse.UploadId,
                };
                completeRequest.AddPartETags(uploadResponses);

                CompleteMultipartUploadResponse completeUploadResponse =
                    await client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (Exception exception)
            {
                Console.WriteLine($"Exception occurred: {exception.Message}");

                // If there was an error, abort the multipart upload.
                AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
                {
```

```
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    UploadId = initResponse.UploadId,
                };

                await client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
    }
```

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez un objet volumineux à l'aide d'un gestionnaire de chargement qui divise les données en plusieurs parties et les charge simultanément.

```
// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3)
 actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform
 bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}



// UploadLargeObject uses an upload manager to upload data to an object in a
 bucket.
```

```
// The upload manager breaks large data into parts and uploads the parts
 concurrently.
func (basics BucketBasics) UploadLargeObject(bucketName string, objectKey string,
 largeObject []byte) error {
 largeBuffer := bytes.NewReader(largeObject)
 var partMiBs int64 = 10
 uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
  u.PartSize = partMiBs * 1024 * 1024
 })
 _, err := uploader.Upload(context.TODO(), &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
  Body:   largeBuffer,
 })
 if err != nil {
  log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }

 return err
}
```

Téléchargez un objet volumineux en utilisant un gestionnaire de téléchargement pour obtenir les données en plusieurs parties et les télécharger simultanément.

```
// DownloadLargeObject uses a download manager to download an object from a
 bucket.
// The download manager gets the data in parts and writes them to a buffer until
 all of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(bucketName string, objectKey
 string) ([]byte, error) {
 var partMiBs int64 = 10
 downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader)
 {
  d.PartSize = partMiBs * 1024 * 1024
 })
 buffer := manager.NewWriteAtBuffer([]byte{})
 _, err := downloader.Download(context.TODO(), buffer, &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
```

```
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return buffer.Bytes(), err
}
```

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Appelez des fonctions qui transfèrent des fichiers vers et depuis un compartiment S3 à l'aide du S3TransferManager.

```
   public Integer downloadObjectsToDirectory(S3TransferManager transferManager,
           URI destinationPathURI, String bucketName) {
       DirectoryDownload directoryDownload =
transferManager.downloadDirectory(DownloadDirectoryRequest.builder()
               .destination(Paths.get(destinationPathURI))
               .bucket(bucketName)
               .build());
       CompletedDirectoryDownload completedDirectoryDownload =
directoryDownload.completionFuture().join();

       completedDirectoryDownload.failedTransfers()
               .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
       return completedDirectoryDownload.failedTransfers().size();
   }
```

Chargez un répertoire local complet.

```
   public Integer uploadDirectory(S3TransferManager transferManager,
           URI sourceDirectory, String bucketName) {
       DirectoryUpload directoryUpload =
transferManager.uploadDirectory(UploadDirectoryRequest.builder()
               .source(Paths.get(sourceDirectory))
               .bucket(bucketName)
               .build());

       CompletedDirectoryUpload completedDirectoryUpload =
directoryUpload.completionFuture().join();
       completedDirectoryUpload.failedTransfers()
               .forEach(fail -> logger.warn("Object [{}] failed to transfer",
fail.toString()));
       return completedDirectoryUpload.failedTransfers().size();
   }
```

Chargez un seul fichier.

```
   public String uploadFile(S3TransferManager transferManager, String
bucketName,
                            String key, URI filePathURI) {
       UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
           .putObjectRequest(b -> b.bucket(bucketName).key(key))
           .source(Paths.get(filePathURI))
           .build();

       FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);

       CompletedFileUpload uploadResult = fileUpload.completionFuture().join();
       return uploadResult.response().eTag();
   }
```

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Chargez un fichier volumineux.

```
import {
  CreateMultipartUploadCommand,
  UploadPartCommand,
  CompleteMultipartUploadCommand,
  AbortMultipartUploadCommand,
  S3Client,
} from "@aws-sdk/client-s3";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

export const main = async () => {
  const s3Client = new S3Client({});
  const bucketName = "test-bucket";
  const key = "multipart.txt";
  const str = createString();
  const buffer = Buffer.from(str, "utf8");

  let uploadId;

  try {
    const multipartUpload = await s3Client.send(
      new CreateMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
      }),
    );
```

```
    uploadId = multipartUpload.UploadId;

    const uploadPromises = [];
    // Multipart uploads require a minimum size of 5 MB per part.
    const partSize = Math.ceil(buffer.length / 5);

    // Upload each part.
    for (let i = 0; i < 5; i++) {
      const start = i * partSize;
      const end = start + partSize;
      uploadPromises.push(
        s3Client
          .send(
            new UploadPartCommand({
              Bucket: bucketName,
              Key: key,
              UploadId: uploadId,
              Body: buffer.subarray(start, end),
              PartNumber: i + 1,
            }),
          )
          .then((d) => {
            console.log("Part", i + 1, "uploaded");
            return d;
          }),
      );
    }

    const uploadResults = await Promise.all(uploadPromises);

    return await s3Client.send(
      new CompleteMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
        UploadId: uploadId,
        MultipartUpload: {
          Parts: uploadResults.map(({ ETag }, i) => ({
            ETag,
            PartNumber: i + 1,
          })),
        },
      }),
    );
```

```
    // Verify the output by downloading the file from the Amazon Simple Storage
 Service (Amazon S3) console.
    // Because the output is a 25 MB string, text editors might struggle to open
 the file.
  } catch (err) {
    console.error(err);

    if (uploadId) {
      const abortCommand = new AbortMultipartUploadCommand({
        Bucket: bucketName,
        Key: key,
        UploadId: uploadId,
      });

      await s3Client.send(abortCommand);
    }
  }
};
```

Téléchargez un fichier volumineux.

```
import { GetObjectCommand, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream } from "fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
```

```
  const [start, end] = range.split("-");
  return {
    start: parseInt(start),
    end: parseInt(end),
    length: parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

// When downloading a large file, you might want to break it down into
// smaller pieces. Amazon S3 accepts a Range header to specify the start
// and end of the byte range to be downloaded.
const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    console.log(`Downloading bytes ${nextRange.start} to ${nextRange.end}`);

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
};

export const main = async () => {
  await downloadInChunks({
    bucket: "my-cool-bucket",
    key: "my-cool-object.txt",
  });
};
```

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez des fonctions qui transfèrent des fichiers en utilisant plusieurs des paramètres disponibles du gestionnaire de transfert. Utilisez une classe de rappel pour écrire la progression des rappels pendant le transfert de fichiers.

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig


MB = 1024 * 1024
s3 = boto3.resource("s3")


class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}
```

```
    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()


def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
```

```
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info


def upload_with_high_threshold(local_file_path, bucket_name, object_key,
 file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
```

```
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey":
 sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
 Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
```

```
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey":
 sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
```

```
            download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
        )
        return transfer_callback.thread_info
```

Faites la démonstration des fonctions du gestionnaire de transfert et établissez un rapport des résultats.

```
import hashlib
import os
import platform
import shutil
import time

import boto3
from boto3.s3.transfer import TransferConfig
from botocore.exceptions import ClientError
from botocore.exceptions import ParamValidationError
from botocore.exceptions import NoCredentialsError

import file_transfer

MB = 1024 * 1024
# These configuration attributes affect both uploads and downloads.
CONFIG_ATTRS = (
    "multipart_threshold",
    "multipart_chunksize",
    "max_concurrency",
    "use_threads",
)
# These configuration attributes affect only downloads.
DOWNLOAD_CONFIG_ATTRS = ("max_io_queue", "io_chunksize", "num_download_attempts")


class TransferDemoManager:
    """
    Manages the demonstration. Collects user input from a command line, reports
    transfer results, maintains a list of artifacts created during the
    demonstration, and cleans them up after the demonstration is completed.
    """
```

```
    def __init__(self):
        self._s3 = boto3.resource("s3")
        self._chore_list = []
        self._create_file_cmd = None
        self._size_multiplier = 0
        self.file_size_mb = 30
        self.demo_folder = None
        self.demo_bucket = None
        self._setup_platform_specific()
        self._terminal_width = shutil.get_terminal_size(fallback=(80, 80))[0]

    def collect_user_info(self):
        """
        Collect local folder and Amazon S3 bucket name from the user. These
        locations are used to store files during the demonstration.
        """
        while not self.demo_folder:
            self.demo_folder = input(
                "Which file folder do you want to use to store " "demonstration
files? "
            )
            if not os.path.isdir(self.demo_folder):
                print(f"{self.demo_folder} isn't a folder!")
                self.demo_folder = None

        while not self.demo_bucket:
            self.demo_bucket = input(
                "Which Amazon S3 bucket do you want to use to store "
                "demonstration files? "
            )
            try:
                self._s3.meta.client.head_bucket(Bucket=self.demo_bucket)
            except ParamValidationError as err:
                print(err)
                self.demo_bucket = None
            except ClientError as err:
                print(err)
                print(
                    f"Either {self.demo_bucket} doesn't exist or you don't "
                    f"have access to it."
                )
                self.demo_bucket = None

    def demo(
```

```
        self, question, upload_func, download_func, upload_args=None,
download_args=None
    ):
        """Run a demonstration.

        Ask the user if they want to run this specific demonstration.
        If they say yes, create a file on the local path, upload it
        using the specified upload function, then download it using the
        specified download function.
        """
        if download_args is None:
            download_args = {}
        if upload_args is None:
            upload_args = {}
        question = question.format(self.file_size_mb)
        answer = input(f"{question} (y/n)")
        if answer.lower() == "y":
            local_file_path, object_key, download_file_path =
self._create_demo_file()

            file_transfer.TransferConfig = self._config_wrapper(
                TransferConfig, CONFIG_ATTRS
            )
            self._report_transfer_params(
                "Uploading", local_file_path, object_key, **upload_args
            )
            start_time = time.perf_counter()
            thread_info = upload_func(
                local_file_path,
                self.demo_bucket,
                object_key,
                self.file_size_mb,
                **upload_args,
            )
            end_time = time.perf_counter()
            self._report_transfer_result(thread_info, end_time - start_time)

            file_transfer.TransferConfig = self._config_wrapper(
                TransferConfig, CONFIG_ATTRS + DOWNLOAD_CONFIG_ATTRS
            )
            self._report_transfer_params(
                "Downloading", object_key, download_file_path, **download_args
            )
            start_time = time.perf_counter()
```

```
            thread_info = download_func(
                self.demo_bucket,
                object_key,
                download_file_path,
                self.file_size_mb,
                **download_args,
            )
            end_time = time.perf_counter()
            self._report_transfer_result(thread_info, end_time - start_time)

    def last_name_set(self):
        """Get the name set used for the last demo."""
        return self._chore_list[-1]

    def cleanup(self):
        """
        Remove files from the demo folder, and uploaded objects from the
        Amazon S3 bucket.
        """
        print("-" * self._terminal_width)
        for local_file_path, s3_object_key, downloaded_file_path in
self._chore_list:
            print(f"Removing {local_file_path}")
            try:
                os.remove(local_file_path)
            except FileNotFoundError as err:
                print(err)

            print(f"Removing {downloaded_file_path}")
            try:
                os.remove(downloaded_file_path)
            except FileNotFoundError as err:
                print(err)

            if self.demo_bucket:
                print(f"Removing {self.demo_bucket}:{s3_object_key}")
                try:

self._s3.Bucket(self.demo_bucket).Object(s3_object_key).delete()
                except ClientError as err:
                    print(err)

    def _setup_platform_specific(self):
        """Set up platform-specific command used to create a large file."""
```

```
        if platform.system() == "Windows":
            self._create_file_cmd = "fsutil file createnew {} {}"
            self._size_multiplier = MB
        elif platform.system() == "Linux" or platform.system() == "Darwin":
            self._create_file_cmd = f"dd if=/dev/urandom of={{}} " f"bs={MB}
count={{}}"
            self._size_multiplier = 1
        else:
            raise EnvironmentError(
                f"Demo of platform {platform.system()} isn't supported."
            )

    def _create_demo_file(self):
        """
        Create a file in the demo folder specified by the user. Store the local
        path, object name, and download path for later cleanup.

        Only the local file is created by this method. The Amazon S3 object and
        download file are created later during the demonstration.

        Returns:
        A tuple that contains the local file path, object name, and download
        file path.
        """
        file_name_template = "TestFile{}-{}.demo"
        local_suffix = "local"
        object_suffix = "s3object"
        download_suffix = "downloaded"
        file_tag = len(self._chore_list) + 1

        local_file_path = os.path.join(
            self.demo_folder, file_name_template.format(file_tag, local_suffix)
        )

        s3_object_key = file_name_template.format(file_tag, object_suffix)

        downloaded_file_path = os.path.join(
            self.demo_folder, file_name_template.format(file_tag,
download_suffix)
        )

        filled_cmd = self._create_file_cmd.format(
            local_file_path, self.file_size_mb * self._size_multiplier
        )
```

```
        print(
            f"Creating file of size {self.file_size_mb} MB "
            f"in {self.demo_folder} by running:"
        )
        print(f"{'':4}{filled_cmd}")
        os.system(filled_cmd)

        chore = (local_file_path, s3_object_key, downloaded_file_path)
        self._chore_list.append(chore)
        return chore

    def _report_transfer_params(self, verb, source_name, dest_name, **kwargs):
        """Report configuration and extra arguments used for a file transfer."""
        print("-" * self._terminal_width)
        print(f"{verb} {source_name} ({self.file_size_mb} MB) to {dest_name}")
        if kwargs:
            print("With extra args:")
            for arg, value in kwargs.items():
                print(f'{"":4}{arg:<20}: {value}')

    @staticmethod
    def ask_user(question):
        """
        Ask the user a yes or no question.

        Returns:
        True when the user answers 'y' or 'Y'; otherwise, False.
        """
        answer = input(f"{question} (y/n) ")
        return answer.lower() == "y"

    @staticmethod
    def _config_wrapper(func, config_attrs):
        def wrapper(*args, **kwargs):
            config = func(*args, **kwargs)
            print("With configuration:")
            for attr in config_attrs:
                print(f'{"":4}{attr:<20}: {getattr(config, attr)}')
            return config

        return wrapper

    @staticmethod
```

```
    def _report_transfer_result(thread_info, elapsed):
        """Report the result of a transfer, including per-thread data."""
        print(f"\nUsed {len(thread_info)} threads.")
        for ident, byte_count in thread_info.items():
            print(f"{'':4}Thread {ident} copied {byte_count} bytes.")
        print(f"Your transfer took {elapsed:.2f} seconds.")


def main():
    """
    Run the demonstration script for s3_file_transfer.
    """
    demo_manager = TransferDemoManager()
    demo_manager.collect_user_info()

    # Upload and download with default configuration. Because the file is 30 MB
    # and the default multipart_threshold is 8 MB, both upload and download are
    # multipart transfers.
    demo_manager.demo(
        "Do you want to upload and download a {} MB file "
        "using the default configuration?",
        file_transfer.upload_with_default_configuration,
        file_transfer.download_with_default_configuration,
    )

    # Upload and download with multipart_threshold set higher than the size of
    # the file. This causes the transfer manager to use standard transfers
    # instead of multipart transfers.
    demo_manager.demo(
        "Do you want to upload and download a {} MB file "
        "as a standard (not multipart) transfer?",
        file_transfer.upload_with_high_threshold,
        file_transfer.download_with_high_threshold,
    )

    # Upload with specific chunk size and additional metadata.
    # Download with a single thread.
    demo_manager.demo(
        "Do you want to upload a {} MB file with a smaller chunk size and "
        "then download the same file using a single thread?",
        file_transfer.upload_with_chunksize_and_meta,
        file_transfer.download_with_single_thread,
        upload_args={
            "metadata": {
```

```
                "upload_type": "chunky",
                "favorite_color": "aqua",
                "size": "medium",
            }
        },
    )

    # Upload using server-side encryption with customer-provided
    # encryption keys.
    # Generate a 256-bit key from a passphrase.
    sse_key = hashlib.sha256("demo_passphrase".encode("utf-8")).digest()
    demo_manager.demo(
        "Do you want to upload and download a {} MB file using "
        "server-side encryption?",
        file_transfer.upload_with_sse,
        file_transfer.download_with_sse,
        upload_args={"sse_key": sse_key},
        download_args={"sse_key": sse_key},
    )

    # Download without specifying an encryption key to show that the
    # encryption key must be included to download an encrypted object.
    if demo_manager.ask_user(
        "Do you want to try to download the encrypted "
        "object without sending the required key?"
    ):
        try:
            _, object_key, download_file_path = demo_manager.last_name_set()
            file_transfer.download_with_default_configuration(
                demo_manager.demo_bucket,
                object_key,
                download_file_path,
                demo_manager.file_size_mb,
            )
        except ClientError as err:
            print(
                "Got expected error when trying to download an encrypted "
                "object without specifying encryption info:"
            )
            print(f"{'':4}{err}")

    # Remove all created and downloaded files, remove all objects from
    # S3 storage.
    if demo_manager.ask_user(
```

```
        "Demonstration complete. Do you want to remove local files " "and S3
 objects?"
    ):
        demo_manager.cleanup()


if __name__ == "__main__":
    try:
        main()
    except NoCredentialsError as error:
        print(error)
        print(
            "To run this example, you must have valid credentials in "
            "a shared credential file or set in environment variables."
        )
```

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
use std::fs::File;
use std::io::prelude::*;
use std::path::Path;

use aws_config::meta::region::RegionProviderChain;
use aws_sdk_s3::error::DisplayErrorContext;
use aws_sdk_s3::operation::{
    create_multipart_upload::CreateMultipartUploadOutput,
 get_object::GetObjectOutput,
};
use aws_sdk_s3::types::{CompletedMultipartUpload, CompletedPart};
use aws_sdk_s3::{config::Region, Client as S3Client};
use aws_smithy_types::byte_stream::{ByteStream, Length};
```

```
use rand::distributions::Alphanumeric;
use rand::{thread_rng, Rng};
use s3_service::error::Error;
use std::process;
use uuid::Uuid;

//In bytes, minimum chunk size of 5MB. Increase CHUNK_SIZE to send larger chunks.
const CHUNK_SIZE: u64 = 1024 * 1024 * 5;
const MAX_CHUNKS: u64 = 10000;

#[tokio::main]
pub async fn main() {
    if let Err(err) = run_example().await {
        eprintln!("Error: {}", DisplayErrorContext(err));
        process::exit(1);
    }
}

async fn run_example() -> Result<(), Error> {
    let shared_config = aws_config::load_from_env().await;
    let client = S3Client::new(&shared_config);

    let bucket_name = format!("doc-example-bucket-{}", Uuid::new_v4());
    let region_provider = RegionProviderChain::first_try(Region::new("us-
west-2"));
    let region = region_provider.region().await.unwrap();
    s3_service::create_bucket(&client, &bucket_name, region.as_ref()).await?;

    let key = "sample.txt".to_string();
    let multipart_upload_res: CreateMultipartUploadOutput = client
        .create_multipart_upload()
        .bucket(&bucket_name)
        .key(&key)
        .send()
        .await
        .unwrap();
    let upload_id = multipart_upload_res.upload_id().unwrap();

    //Create a file of random characters for the upload.
    let mut file = File::create(&key).expect("Could not create sample file.");
    // Loop until the file is 5 chunks.
    while file.metadata().unwrap().len() <= CHUNK_SIZE * 4 {
        let rand_string: String = thread_rng()
            .sample_iter(&Alphanumeric)
```

```
            .take(256)
            .map(char::from)
            .collect();
        let return_string: String = "\n".to_string();
        file.write_all(rand_string.as_ref())
            .expect("Error writing to file.");
        file.write_all(return_string.as_ref())
            .expect("Error writing to file.");
    }

    let path = Path::new(&key);
    let file_size = tokio::fs::metadata(path)
        .await
        .expect("it exists I swear")
        .len();

    let mut chunk_count = (file_size / CHUNK_SIZE) + 1;
    let mut size_of_last_chunk = file_size % CHUNK_SIZE;
    if size_of_last_chunk == 0 {
        size_of_last_chunk = CHUNK_SIZE;
        chunk_count -= 1;
    }

    if file_size == 0 {
        panic!("Bad file size.");
    }
    if chunk_count > MAX_CHUNKS {
        panic!("Too many chunks! Try increasing your chunk size.")
    }

    let mut upload_parts: Vec<CompletedPart> = Vec::new();

    for chunk_index in 0..chunk_count {
        let this_chunk = if chunk_count - 1 == chunk_index {
            size_of_last_chunk
        } else {
            CHUNK_SIZE
        };
        let stream = ByteStream::read_from()
            .path(path)
            .offset(chunk_index * CHUNK_SIZE)
            .length(Length::Exact(this_chunk))
            .build()
            .await
```

```
            .unwrap();
        //Chunk index needs to start at 0, but part numbers start at 1.
        let part_number = (chunk_index as i32) + 1;
        let upload_part_res = client
            .upload_part()
            .key(&key)
            .bucket(&bucket_name)
            .upload_id(upload_id)
            .body(stream)
            .part_number(part_number)
            .send()
            .await?;
        upload_parts.push(
            CompletedPart::builder()
                .e_tag(upload_part_res.e_tag.unwrap_or_default())
                .part_number(part_number)
                .build(),
        );
    }
    let completed_multipart_upload: CompletedMultipartUpload =
CompletedMultipartUpload::builder()
        .set_parts(Some(upload_parts))
        .build();

    let _complete_multipart_upload_res = client
        .complete_multipart_upload()
        .bucket(&bucket_name)
        .key(&key)
        .multipart_upload(completed_multipart_upload)
        .upload_id(upload_id)
        .send()
        .await
        .unwrap();

    let data: GetObjectOutput = s3_service::download_object(&client,
&bucket_name, &key).await?;
    let data_length: u64 = data
        .content_length()
        .unwrap_or_default()
        .try_into()
        .unwrap();
    if file.metadata().unwrap().len() == data_length {
        println!("Data lengths match.");
    } else {
```

```
        println!("The data was not the same size!");
    }

    s3_service::delete_objects(&client, &bucket_name)
        .await
        .expect("Error emptying bucket.");
    s3_service::delete_bucket(&client, &bucket_name)
        .await
        .expect("Error deleting bucket.");

    Ok(())
}
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Chargez un flux de taille inconnue vers un objet Amazon S3 à l'aide d'un AWS SDK

L'exemple de code suivant montre comment charger un flux de taille inconnue dans un objet Amazon S3.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Utilisez le Client S3 basé sur CRT AWS.

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
```

```
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.services.s3.S3AsyncClient;
import software.amazon.awssdk.services.s3.model.PutObjectResponse;

import java.io.ByteArrayInputStream;
import java.util.UUID;
import java.util.concurrent.CompletableFuture;

    /**
     * @param s33CrtAsyncClient - To upload content from a stream of unknown
 size, use the AWS CRT-based S3 client. For more information, see
     *                          https://docs.aws.amazon.com/sdk-for-java/latest/
developer-guide/crt-based-s3-client.html.
     * @param bucketName - The name of the bucket.
     * @param key - The name of the object.
     * @return software.amazon.awssdk.services.s3.model.PutObjectResponse -
 Returns metadata pertaining to the put object operation.
     */
    public PutObjectResponse putObjectFromStream(S3AsyncClient s33CrtAsyncClient,
 String bucketName, String key) {

        BlockingInputStreamAsyncRequestBody body =
                AsyncRequestBody.forBlockingInputStream(null); // 'null'
 indicates a stream will be provided later.

        CompletableFuture<PutObjectResponse> responseFuture =
                s33CrtAsyncClient.putObject(r -> r.bucket(bucketName).key(key),
 body);

        // AsyncExampleUtils.randomString() returns a random string up to 100
 characters.
        String randomString = AsyncExampleUtils.randomString();
        logger.info("random string to upload: {}: length={}", randomString,
 randomString.length());

        // Provide the stream of data to be uploaded.
        body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

        PutObjectResponse response = responseFuture.join(); // Wait for the
 response.
        logger.info("Object {} uploaded to bucket {}.", key, bucketName);
        return response;
    }
```

```
}
```

Utilisez le Gestionnaire de transferts Amazon S3.

```
import com.example.s3.util.AsyncExampleUtils;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.async.AsyncRequestBody;
import software.amazon.awssdk.core.async.BlockingInputStreamAsyncRequestBody;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.CompletedUpload;
import software.amazon.awssdk.transfer.s3.model.Upload;

import java.io.ByteArrayInputStream;
import java.util.UUID;

    /**
     * @param transferManager - To upload content from a stream of unknown size,
 use the S3TransferManager based on the AWS CRT-based S3 client.
     *                        For more information, see https://
docs.aws.amazon.com/sdk-for-java/latest/developer-guide/transfer-manager.html.
     * @param bucketName - The name of the bucket.
     * @param key - The name of the object.
     * @return - software.amazon.awssdk.transfer.s3.model.CompletedUpload - The
 result of the completed upload.
     */
    public CompletedUpload uploadStream(S3TransferManager transferManager, String
 bucketName, String key) {

        BlockingInputStreamAsyncRequestBody body =
                AsyncRequestBody.forBlockingInputStream(null); // 'null'
 indicates a stream will be provided later.

        Upload upload = transferManager.upload(builder -> builder
                .requestBody(body)
                .putObjectRequest(req -> req.bucket(bucketName).key(key))
                .build());

        // AsyncExampleUtils.randomString() returns a random string up to 100
 characters.
        String randomString = AsyncExampleUtils.randomString();
```

```
        logger.info("random string to upload: {}: length={}", randomString,
 randomString.length());

        // Provide the stream of data to be uploaded.
        body.writeInputStream(new ByteArrayInputStream(randomString.getBytes()));

        return upload.completionFuture().join();
    }
}
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisez des checksums pour travailler avec un objet Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant montre comment utiliser des sommes de contrôle pour travailler avec un objet Amazon S3.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Les exemples de code utilisent un sous-ensemble des importations suivantes.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.exception.SdkException;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
```

```
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.UploadPartRequest;
import software.amazon.awssdk.services.s3.model.UploadPartResponse;
import software.amazon.awssdk.services.s3.waiters.S3Waiter;
import software.amazon.awssdk.transfer.s3.S3TransferManager;
import software.amazon.awssdk.transfer.s3.model.FileUpload;
import software.amazon.awssdk.transfer.s3.model.UploadFileRequest;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.net.URISyntaxException;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.file.Paths;
import java.security.DigestInputStream;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.List;
import java.util.Objects;
import java.util.UUID;
```

Spécifiez un algorithme de somme de contrôle pour la méthode `putObject` lorsque vous créez l'élément `PutObjectRequest`.

```
    public void putObjectWithChecksum() {
        s3Client.putObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumAlgorithm(ChecksumAlgorithm.CRC32),
                RequestBody.fromString("This is a test"));
    }
```

Vérifiez la somme de contrôle de la `getObject` méthode lorsque vous créez le GetObjectRequest.

```
    public GetObjectResponse getObjectWithChecksum() {
        return s3Client.getObject(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumMode(ChecksumMode.ENABLED))
                .response();
    }
```

Pré-calculez une somme de contrôle pour la méthode `putObject` lorsque vous créez l'élément `PutObjectRequest`.

```
    public void putObjectWithPrecalculatedChecksum(String filePath) {
        String checksum = calculateChecksum(filePath, "SHA-256");

        s3Client.putObject((b -> b
                .bucket(bucketName)
                .key(key)
                .checksumSHA256(checksum)),
                RequestBody.fromFile(Paths.get(filePath)));
    }
```

Utilisez le gestionnaire de transferts S3 situé au-dessus du client S3 basé sur AWS CRT pour effectuer de manière transparente un téléchargement partitionné lorsque la taille du contenu dépasse un seuil. La taille par défaut est de 8 Mo.

Vous pouvez spécifier un algorithme de somme de contrôle que le kit SDK utilisera. Par défaut, le kit SDK utilise l'algorithme CRC32.

```
    public void multipartUploadWithChecksumTm(String filePath) {
        S3TransferManager transferManager = S3TransferManager.create();
        UploadFileRequest uploadFileRequest = UploadFileRequest.builder()
                .putObjectRequest(b -> b
                        .bucket(bucketName)
                        .key(key)
                        .checksumAlgorithm(ChecksumAlgorithm.SHA1))
                .source(Paths.get(filePath))
                .build();
        FileUpload fileUpload = transferManager.uploadFile(uploadFileRequest);
        fileUpload.completionFuture().join();
```

```
        transferManager.close();
    }
```

Utilisez l'API S3Client ou (AsyncClient API S3) pour effectuer un téléchargement partitionné. Si vous spécifiez une somme de contrôle supplémentaire, vous devez spécifier l'algorithme à utiliser lors du lancement du chargement. Vous devez également spécifier l'algorithme pour chaque demande d'article et fournir la somme de contrôle calculée pour chaque article après son chargement.

```
    public void multipartUploadWithChecksumS3Client(String filePath) {
        ChecksumAlgorithm algorithm = ChecksumAlgorithm.CRC32;

        // Initiate the multipart upload.
        CreateMultipartUploadResponse createMultipartUploadResponse =
s3Client.createMultipartUpload(b -> b
                .bucket(bucketName)
                .key(key)
                .checksumAlgorithm(algorithm)); // Checksum specified on
initiation.
        String uploadId = createMultipartUploadResponse.uploadId();

        // Upload the parts of the file.
        int partNumber = 1;
        List<CompletedPart> completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            long position = 0;
            while (position < fileSize) {
                file.seek(position);
                long read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .checksumAlgorithm(algorithm) // Checksum specified on
each part.
                        .partNumber(partNumber)
```

```
                        .build();

                UploadPartResponse partResponse = s3Client.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .checksumCRC32(partResponse.checksumCRC32()) // Provide
the calculated checksum.
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
                position += read;
                partNumber++;
            }
        } catch (IOException e) {
            System.err.println(e.getMessage());
        }

        // Complete the multipart upload.
        s3Client.completeMultipartUpload(b -> b
                .bucket(bucketName)
                .key(key)
                .uploadId(uploadId)

.multipartUpload(CompletedMultipartUpload.builder().parts(completedParts).build()));
    }
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - CompleteMultipartUpload
  - CreateMultipartUpload
  - UploadPart

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Travaillez avec des objets versionnés Amazon S3 à l'aide d'un SDK AWS

L'exemple de code suivant illustre comment :

- Créez un compartiment S3 versionné.
- Obtenez toutes les versions d'un objet.
- Rétablissez un objet à une version précédente.
- Supprimez et restaurez un objet versionné.
- Supprimez définitivement toutes les versions d'un objet.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez des fonctions qui enveloppent les actions S3.

```
def create_versioned_bucket(bucket_name, prefix):
    """
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
 lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
```

```
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
```

```
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )

    return bucket



def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
```

```
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
            else:
                break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
 f"{object_key}."
        )



def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
    A versioned object presents as deleted when its latest version is a delete
 marker.
    By removing the delete marker, we make the previous version the latest
 version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
                latest_version["LastModified"],
            )
```

```
            obj = bucket.Object(object_key)
            obj.Version(latest_version["VersionId"]).delete()
            logger.info(
                "Revived %s, active version is now %s  with body '%s'",
                object_key,
                obj.version_id,
                obj.get()["Body"].read(),
            )
        else:
            logger.warning(
                "Delete marker is not the latest version for %s!", object_key
            )
    elif "Versions" in response:
        logger.warning("Got an active version for %s, nothing to do.",
 object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)



def permanently_delete_object(bucket, object_key):
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Chargez la strophe d'un poème dans un objet versionné et effectuez une série d'actions sur celle-ci.

```
def usage_demo_single_object(obj_prefix="demo-versioning/"):
    """
    Demonstrates usage of versioned object functions. This demo uploads a stanza
    of a poem and performs a series of revisions, deletions, and revivals on it.

    :param obj_prefix: The prefix to assign to objects created by this demo.
    """
    with open("father_william.txt") as file:
        stanzas = file.read().split("\n\n")

    width = get_terminal_size((80, 20))[0]
    print("-" * width)
    print("Welcome to the usage demonstration of Amazon S3 versioning.")
    print(
        "This demonstration uploads a single stanza of a poem to an Amazon "
        "S3 bucket and then applies various revisions to it."
    )
    print("-" * width)
    print("Creating a version-enabled bucket for the demo...")
    bucket = create_versioned_bucket("bucket-" + str(uuid.uuid1()), obj_prefix)

    print("\nThe initial version of our stanza:")
    print(stanzas[0])

    # Add the first stanza and revise it a few times.
    print("\nApplying some revisions to the stanza...")
    obj_stanza_1 = bucket.Object(f"{obj_prefix}stanza-1")
    obj_stanza_1.put(Body=bytes(stanzas[0], "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0].upper(), "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0].lower(), "utf-8"))
    obj_stanza_1.put(Body=bytes(stanzas[0][::-1], "utf-8"))
    print(
        "The latest version of the stanza is now:",
        obj_stanza_1.get()["Body"].read().decode("utf-8"),
        sep="\n",
    )

    # Versions are returned in order, most recent first.
    obj_stanza_1_versions =
 bucket.object_versions.filter(Prefix=obj_stanza_1.key)
    print(
        "The version data of the stanza revisions:",
        *[
```

```
            f"    {version.version_id}, last modified {version.last_modified}"
            for version in obj_stanza_1_versions
        ],
        sep="\n",
    )

    # Rollback two versions.
    print("\nRolling back two versions...")
    rollback_object(bucket, obj_stanza_1.key, list(obj_stanza_1_versions)
[2].version_id)
    print(
        "The latest version of the stanza:",
        obj_stanza_1.get()["Body"].read().decode("utf-8"),
        sep="\n",
    )

    # Delete the stanza
    print("\nDeleting the stanza...")
    obj_stanza_1.delete()
    try:
        obj_stanza_1.get()
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchKey":
            print("The stanza is now deleted (as expected).")
        else:
            raise

    # Revive the stanza
    print("\nRestoring the stanza...")
    revive_object(bucket, obj_stanza_1.key)
    print(
        "The stanza is restored! The latest version is again:",
        obj_stanza_1.get()["Body"].read().decode("utf-8"),
        sep="\n",
    )

    # Permanently delete all versions of the object. This cannot be undone!
    print("\nPermanently deleting all versions of the stanza...")
    permanently_delete_object(bucket, obj_stanza_1.key)
    obj_stanza_1_versions =
 bucket.object_versions.filter(Prefix=obj_stanza_1.key)
    if len(list(obj_stanza_1_versions)) == 0:
        print("The stanza has been permanently deleted and now has no versions.")
    else:
```

```
            print("Something went wrong. The stanza still exists!")

    print(f"\nRemoving {bucket.name}...")
    bucket.delete()
    print(f"{bucket.name} deleted.")
    print("Demo done!")
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - CreateBucket
  - DeleteObject
  - ListObjectVersions
  - PutBucketLifecycleConfiguration

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples de solutions sans serveur pour Amazon S3 utilisant des SDK AWS

Les exemples de code suivants montrent comment utiliser Amazon S3 avec des AWS SDK.

Exemples

- Invoquer une fonction lambda à partir d'un déclencheur Amazon S3

Invoquer une fonction lambda à partir d'un déclencheur Amazon S3

Les exemples de code suivants montrent comment implémenter une fonction Lambda qui reçoit un événement déclenché par le chargement d'un objet vers un compartiment S3. La fonction extrait le nom du compartiment S3 et la clé de l'objet à partir du paramètre d'événement et appelle l'API Amazon S3 pour récupérer et consigner le type de contenu de l'objet.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Utilisation d'un événement S3 avec Lambda en utilisant .NET.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
using System.Threading.Tasks;
using Amazon.Lambda.Core;
using Amazon.S3;
using System;
using Amazon.Lambda.S3Events;
using System.Web;

// Assembly attribute to enable the Lambda function's JSON input to be converted
 into a .NET class.
[assembly:
 LambdaSerializer(typeof(Amazon.Lambda.Serialization.SystemTextJson.DefaultLambdaJsonSeri

namespace S3Integration
{
    public class Function
    {
        private static AmazonS3Client _s3Client;
        public Function() : this(null)
        {
        }

        internal Function(AmazonS3Client s3Client)
        {
            _s3Client = s3Client ?? new AmazonS3Client();
        }

        public async Task<string> Handler(S3Event evt, ILambdaContext context)
        {
            try
```

```
            {
                if (evt.Records.Count <= 0)
                {
                    context.Logger.LogLine("Empty S3 Event received");
                    return string.Empty;
                }

                var bucket = evt.Records[0].S3.Bucket.Name;
                var key = HttpUtility.UrlDecode(evt.Records[0].S3.Object.Key);

                context.Logger.LogLine($"Request is for {bucket} and {key}");

                var objectResult = await _s3Client.GetObjectAsync(bucket, key);

                context.Logger.LogLine($"Returning {objectResult.Key}");

                return objectResult.Key;
            }
            catch (Exception e)
            {
                context.Logger.LogLine($"Error processing request -
 {e.Message}");

                return string.Empty;
            }
        }
    }
}
```

Go

Kit SDK for Go V2

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Utilisation d'un événement S3 avec Lambda en utilisant Go.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
// SPDX-License-Identifier: Apache-2.0
package main

import (
 "context"
 "log"

 "github.com/aws/aws-lambda-go/events"
 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/s3"
)

func handler(ctx context.Context, s3Event events.S3Event) error {
 sdkConfig, err := config.LoadDefaultConfig(ctx)
 if err != nil {
  log.Printf("failed to load default config: %s", err)
  return err
 }
 s3Client := s3.NewFromConfig(sdkConfig)

 for _, record := range s3Event.Records {
  bucket := record.S3.Bucket.Name
  key := record.S3.Object.URLDecodedKey
  headOutput, err := s3Client.HeadObject(ctx, &s3.HeadObjectInput{
   Bucket: &bucket,
   Key:    &key,
  })
  if err != nil {
   log.Printf("error getting head of object %s/%s: %s", bucket, key, err)
   return err
  }
  log.Printf("successfully retrieved %s/%s of type %s", bucket, key,
 *headOutput.ContentType)
 }

 return nil
}

func main() {
 lambda.Start(handler)
}
```

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Utilisation d'un événement S3 avec Lambda en utilisant Go.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
package example;

import software.amazon.awssdk.services.s3.model.HeadObjectRequest;
import software.amazon.awssdk.services.s3.model.HeadObjectResponse;
import software.amazon.awssdk.services.s3.S3Client;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;
import com.amazonaws.services.lambda.runtime.events.S3Event;
import
 com.amazonaws.services.lambda.runtime.events.models.s3.S3EventNotification.S3EventNotifi

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Handler implements RequestHandler<S3Event, String> {
    private static final Logger logger = LoggerFactory.getLogger(Handler.class);
    @Override
    public String handleRequest(S3Event s3event, Context context) {
        try {
          S3EventNotificationRecord record = s3event.getRecords().get(0);
          String srcBucket = record.getS3().getBucket().getName();
          String srcKey = record.getS3().getObject().getUrlDecodedKey();

          S3Client s3Client = S3Client.builder().build();
          HeadObjectResponse headObject = getHeadObject(s3Client, srcBucket,
 srcKey);
```

```
            logger.info("Successfully retrieved " + srcBucket + "/" + srcKey + " of
 type " + headObject.contentType());

            return "Ok";
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }

    private HeadObjectResponse getHeadObject(S3Client s3Client, String bucket,
 String key) {
        HeadObjectRequest headObjectRequest = HeadObjectRequest.builder()
                .bucket(bucket)
                .key(key)
                .build();
        return s3Client.headObject(headObjectRequest);
    }
}
```

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Consommation d'un événement S3 avec Lambda en utilisant. JavaScript

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { S3Client, HeadObjectCommand } from "@aws-sdk/client-s3";

const client = new S3Client();

exports.handler = async (event, context) => {

    // Get the object from the event and show its content type
    const bucket = event.Records[0].s3.bucket.name;
```

```
    const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g,
 ' '));

    try {
        const { ContentType } = await client.send(new HeadObjectCommand({
            Bucket: bucket,
            Key: key,
        }));

        console.log('CONTENT TYPE:', ContentType);
        return ContentType;

    } catch (err) {
        console.log(err);
        const message = `Error getting object ${key} from bucket ${bucket}. Make
 sure they exist and your bucket is in the same region as this function.`;
        console.log(message);
        throw new Error(message);
    }
};
```

Consommation d'un événement S3 avec Lambda en utilisant. TypeScript

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
import { S3Event } from 'aws-lambda';
import { S3Client, HeadObjectCommand } from '@aws-sdk/client-s3';

const s3 = new S3Client({ region: process.env.AWS_REGION });

export const handler = async (event: S3Event): Promise<string | undefined> => {
  // Get the object from the event and show its content type
  const bucket = event.Records[0].s3.bucket.name;
  const key = decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g, '
 '));
  const params = {
    Bucket: bucket,
    Key: key,
  };
  try {
    const { ContentType } = await s3.send(new HeadObjectCommand(params));
    console.log('CONTENT TYPE:', ContentType);
```

```
    return ContentType;
  } catch (err) {
    console.log(err);
    const message = `Error getting object ${key} from bucket ${bucket}. Make sure
 they exist and your bucket is in the same region as this function.`;
    console.log(message);
    throw new Error(message);
  }
};
```

PHP

Kit SDK pour PHP

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Consommation d'un événement S3 avec Lambda à l'aide de PHP.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
<?php

use Bref\Context\Context;
use Bref\Event\S3\S3Event;
use Bref\Event\S3\S3Handler;
use Bref\Logger\StderrLogger;

require __DIR__ . '/vendor/autoload.php';


class Handler extends S3Handler
{
    private StderrLogger $logger;
    public function __construct(StderrLogger $logger)
    {
        $this->logger = $logger;
    }
```

```
    public function handleS3(S3Event $event, Context $context) : void
    {
        $this->logger->info("Processing S3 records");

        // Get the object from the event and show its content type
        $records = $event->getRecords();

        foreach ($records as $record)
        {
            $bucket = $record->getBucket()->getName();
            $key = urldecode($record->getObject()->getKey());

            try {
                $fileSize = urldecode($record->getObject()->getSize());
                echo "File Size: " . $fileSize . "\n";
                // TODO: Implement your custom processing logic here
            } catch (Exception $e) {
                echo $e->getMessage() . "\n";
                echo 'Error getting object ' . $key . ' from bucket ' .
 $bucket . '. Make sure they exist and your bucket is in the same region as this
 function.' . "\n";
                throw $e;
            }
        }
    }
}

$logger = new StderrLogger();
return new Handler($logger);
```

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Utilisation d'un événement S3 avec Lambda en utilisant Python.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
import json
import urllib.parse
import boto3

print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # Get the object from the event and show its content type
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'],
 encoding='utf-8')
    try:
        response = s3.get_object(Bucket=bucket, Key=key)
        print("CONTENT TYPE: " + response['ContentType'])
        return response['ContentType']
    except Exception as e:
        print(e)
        print('Error getting object {} from bucket {}. Make sure they exist and
 your bucket is in the same region as this function.'.format(key, bucket))
        raise e
```

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Consommation d'un événement S3 avec Lambda à l'aide de Ruby.

```
require 'json'
require 'uri'
require 'aws-sdk'

puts 'Loading function'

def lambda_handler(event:, context:)
  s3 = Aws::S3::Client.new(region: 'region') # Your AWS region
  # puts "Received event: #{JSON.dump(event)}"

  # Get the object from the event and show its content type
  bucket = event['Records'][0]['s3']['bucket']['name']
  key = URI.decode_www_form_component(event['Records'][0]['s3']['object']['key'],
 Encoding::UTF_8)
  begin
    response = s3.get_object(bucket: bucket, key: key)
    puts "CONTENT TYPE: #{response.content_type}"
    return response.content_type
  rescue StandardError => e
    puts e.message
    puts "Error getting object #{key} from bucket #{bucket}. Make sure they exist
 and your bucket is in the same region as this function."
    raise e
  end
end
```

Rust

SDK pour Rust

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples sans serveur.

Utilisation d'un événement S3 avec Lambda en utilisant Rust.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
```

```
use aws_lambda_events::event::s3::S3Event;
use aws_sdk_s3::{Client};
use lambda_runtime::{run, service_fn, Error, LambdaEvent};


/// Main function
#[tokio::main]
async fn main() -> Result<(), Error> {
    tracing_subscriber::fmt()
        .with_max_level(tracing::Level::INFO)
        .with_target(false)
        .without_time()
        .init();

    // Initialize the AWS SDK for Rust
    let config = aws_config::load_from_env().await;
    let s3_client = Client::new(&config);

    let res = run(service_fn(|request: LambdaEvent<S3Event>| {
        function_handler(&s3_client, request)
    })).await;

    res
}

async fn function_handler(
    s3_client: &Client,
    evt: LambdaEvent<S3Event>
) -> Result<(), Error> {
    tracing::info!(records = ?evt.payload.records.len(), "Received request from
 SQS");

    if evt.payload.records.len() == 0 {
        tracing::info!("Empty S3 event received");
    }

    let bucket = evt.payload.records[0].s3.bucket.name.as_ref().expect("Bucket
 name to exist");
    let key = evt.payload.records[0].s3.object.key.as_ref().expect("Object key to
 exist");

    tracing::info!("Request is for {} and object {}", bucket, key);

    let s3_get_object_result = s3_client
```

```
        .get_object()
        .bucket(bucket)
        .key(key)
        .send()
        .await;

    match s3_get_object_result {
        Ok(_) => tracing::info!("S3 Get Object success, the s3GetObjectResult
 contains a 'body' property of type ByteStream"),
        Err(_) => tracing::info!("Failure with S3 Get Object request")
    }

    Ok(())
}
```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples multiservices pour Amazon S3 utilisant AWS des kits de développement logiciel

Les exemples d'applications suivants utilisent des AWS SDK pour combiner Amazon S3 avec d'autres Services AWS applications. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter l'application.

Exemples

- Créer une application Amazon Transcribe
- Convertissez du texte en parole et de nouveau en texte à l'aide d'un AWS SDK
- Création d’une application de gestion des ressources photographiques permettant aux utilisateurs de gérer les photos à l’aide d’étiquettes
- Créer une application Amazon Textract Explorer
- Détectez le PPE dans les images avec Amazon Rekognition à l'aide d'un SDK AWS
- Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK
- Détecter les visages dans une image à l'aide d'un AWS SDK
- Détectez des objets dans des images avec Amazon Rekognition à l'aide d'un SDK AWS

- Détectez les personnes et les objets dans une vidéo avec Amazon Rekognition à l'aide d'un SDK AWS
- Enregistrez les informations EXIF et autres informations sur les images à l'aide d'un SDK AWS
- Transformez les données de votre application avec S3 Object Lambda

Créer une application Amazon Transcribe

L'exemple de code suivant montre comment utiliser Amazon Transcribe pour transcrire et afficher des enregistrements vocaux dans le navigateur.

JavaScript

SDK pour JavaScript (v3)

Créez une application qui utilise Amazon Transcribe pour transcrire et afficher des enregistrements vocaux dans le navigateur. L'application utilise deux compartiments Amazon Simple Storage Service (Amazon S3), l'un pour héberger le code de l'application, l'autre pour stocker les transcriptions. L'application utilise un groupe d'utilisateurs Amazon Cognito pour authentifier vos utilisateurs. Les utilisateurs authentifiés disposent des autorisations AWS Identity and Access Management (IAM) nécessaires pour accéder aux services requis. AWS

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Cet exemple est également disponible dans le AWS SDK for JavaScript guide du développeur v3.

Les services utilisés dans cet exemple

- Amazon Cognito Identity
- Amazon S3
- Amazon Transcribe

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Convertissez du texte en parole et de nouveau en texte à l'aide d'un AWS SDK

L’exemple de code suivant illustre comment :

- Utilisez Amazon Polly pour synthétiser un fichier d’entrée en texte brut (UTF-8) en un fichier audio.
- Chargez le fichier audio sur un compartiment Amazon S3.
- Utilisez Amazon Transcribe pour convertir le fichier audio en texte.
- Affichez le texte.

Rust

SDK pour Rust

Utilisez Amazon Polly pour synthétiser un fichier d’entrée en texte brut (UTF-8) en un fichier audio, charger le fichier audio dans un compartiment Amazon S3, utiliser Amazon Transcribe pour convertir ce fichier audio en texte et afficher le texte.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Polly
- Amazon S3
- Amazon Transcribe

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Création d’une application de gestion des ressources photographiques permettant aux utilisateurs de gérer les photos à l’aide d’étiquettes

Les exemples de code suivants montrent comment créer une application sans serveur permettant aux utilisateurs de gérer des photos à l’aide d’étiquettes.

.NET

AWS SDK for .NET

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

C++

SDK pour C++

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda

- Amazon Rekognition
- Amazon S3
- Amazon SNS

Java

SDK pour Java 2.x

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

JavaScript

SDK pour JavaScript (v3)

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Kotlin

SDK pour Kotlin

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

PHP

Kit SDK pour PHP

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Rust

SDK pour Rust

Montre comment développer une application de gestion de ressources photographiques qui détecte les étiquettes dans les images à l'aide d'Amazon Rekognition et les stocke pour les récupérer ultérieurement.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Pour explorer en profondeur l'origine de cet exemple, consultez l'article sur AWS Community.

Les services utilisés dans cet exemple

- API Gateway
- DynamoDB
- Lambda
- Amazon Rekognition
- Amazon S3
- Amazon SNS

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Créer une application Amazon Textract Explorer

Les exemples de code suivants expliquent comment explorer la sortie Amazon Textract via une application interactive.

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser le AWS SDK for JavaScript pour créer une application React qui utilise Amazon Textract pour extraire des données d'une image de document et les afficher sur une page Web interactive. Cet exemple s'exécute dans un navigateur Web et nécessite une identité Amazon Cognito authentifiée pour les informations d'identification. Il utilise Amazon Simple Storage Service (Amazon S3) pour le stockage et, pour les notifications, il interroge une file d'attente Amazon Simple Queue Service (Amazon SQS) abonnée à une rubrique Amazon Simple Notification Service (Amazon SNS).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Cognito Identity
- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Python

SDK pour Python (Boto3)

Montre comment utiliser Amazon Textract pour détecter des éléments de texte, de formulaire et de tableau dans une image de document. AWS SDK for Python (Boto3) L'image d'entrée

et la sortie d'Amazon Textract sont affichées dans une application Tkinter qui vous permet d'explorer les éléments détectés.

- Soumettez une image de document à Amazon Textract et explorez la sortie des éléments détectés.
- Soumettez des images directement à Amazon Textract ou via un compartiment Amazon Simple Storage Service (Amazon S3).
- Utilisez des API asynchrones pour démarrer une tâche qui publie une notification dans une rubrique Amazon Simple Notification Service (Amazon SNS) lorsque le travail est terminé.
- Interrogez un service Amazon Simple Queue Service (Amazon SQS) pour obtenir un message de fin de tâche et affichez les résultats.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon S3
- Amazon SNS
- Amazon SQS
- Amazon Textract

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détectez le PPE dans les images avec Amazon Rekognition à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment créer une application qui utilise Amazon Rekognition afin de détecter l'équipement de protection individuelle (EPI) dans les images.

Java

SDK pour Java 2.x

Montre comment créer une AWS Lambda fonction qui détecte les images à l'aide d'un équipement de protection individuelle.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser Amazon Rekognition AWS SDK for JavaScript pour créer une application permettant de détecter les équipements de protection individuelle (EPI) sur des images situées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application enregistre les résultats dans une table Amazon DynamoDB et envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Découvrez comment :

- Créer un utilisateur non authentifié à l'aide d'Amazon Cognito.
- Analyser les images à la recherche d'EPI à l'aide d'Amazon Rekognition.
- Vérifier une adresse e-mail pour Amazon SES.
- Mettre à jour une table DynamoDB avec les résultats.
- Envoyer une notification par e-mail à l'aide d'Amazon SES.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- DynamoDB
- Amazon Rekognition
- Amazon S3
- Amazon SES

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détecter les entités dans le texte extrait d'une image à l'aide d'un AWS SDK

L'exemple de code suivant montre comment utiliser Amazon Comprehend pour détecter des entités dans du texte extrait par Amazon Textract à partir d'une image stockée dans Amazon S3.

Python

SDK pour Python (Boto3)

Montre comment utiliser le AWS SDK for Python (Boto3) dans un bloc-notes Jupyter pour détecter des entités dans du texte extrait d'une image. Cet exemple utilise Amazon Textract pour extraire le texte d'une image stockée dans Amazon Simple Storage Service (Amazon S3) et Amazon Comprehend pour détecter les entités dans le texte extrait.

Cet exemple est un carnet Jupyter et doit être exécuté dans un environnement qui peut accueillir des carnets. Pour savoir comment exécuter cet exemple à l'aide d'Amazon SageMaker, consultez les instructions du TextractAndComprehendNotebookfichier .ipynb.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Comprehend
- Amazon S3
- Amazon Textract

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détecter les visages dans une image à l'aide d'un AWS SDK

L’exemple de code suivant illustre comment :

- Enregistrez une image dans un compartiment Amazon S3.

- Utilisez Amazon Rekognition pour détecter les détails du visage, tels que la tranche d'âge, le sexe et l'émotion (sourire, etc.).
- Affichez ces détails.

Rust

SDK pour Rust

Enregistrez l'image dans un compartiment Amazon S3 avec un préfixe uploads (chargement), utilisez Amazon Rekognition pour détecter les détails du visage, tels que la tranche d'âge, le sexe et l'émotion (sourire, etc.) et affichez ces détails.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détectez des objets dans des images avec Amazon Rekognition à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment créer une application qui utilise Amazon Rekognition afin de détecter des objets par catégorie dans des images.

.NET

AWS SDK for .NET

Montre comment utiliser l'API Java Amazon Rekognition afin de créer une application qui, avec Amazon Rekognition, permet d'identifier des objets par catégorie dans des images stockées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

Java

SDK pour Java 2.x

Montre comment utiliser l'API Java Amazon Rekognition afin de créer une application qui, avec Amazon Rekognition, permet d'identifier des objets par catégorie dans des images stockées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser Amazon Rekognition AWS SDK for JavaScript pour créer une application qui utilise Amazon Rekognition pour identifier les objets par catégorie dans des images situées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Découvrez comment :

- Créer un utilisateur non authentifié à l'aide d'Amazon Cognito.
- Analyser les images à la recherche d'objets à l'aide d'Amazon Rekognition.
- Vérifier une adresse e-mail pour Amazon SES.
- Envoyer une notification par e-mail à l'aide d'Amazon SES.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

Kotlin

SDK pour Kotlin

Montre comment utiliser l'API Kotlin Amazon Rekognition afin de créer une application qui, avec Amazon Rekognition, permet d'identifier des objets par catégorie dans des images stockées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

Python

SDK pour Python (Boto3)

Vous montre comment utiliser le AWS SDK for Python (Boto3) pour créer une application Web qui vous permet d'effectuer les opérations suivantes :

- Chargez les photos dans un compartiment Amazon Simple Storage Service (Amazon S3).
- Utilisez Amazon Rekognition pour analyser et étiqueter les photos.
- Utilisez Amazon Simple Email Service (Amazon SES) pour envoyer des rapports d'analyse d'images par e-mail.

Cet exemple contient deux composants principaux : une page Web écrite avec React et un service REST écrit en Python construit avec Flask-RESTful. JavaScript

Vous pouvez utiliser la page web React pour :

- Affichez une liste d'images stockées dans votre compartiment S3.
- Chargez des images depuis votre ordinateur dans votre compartiment S3.
- Affichez des images et des étiquettes qui identifient les éléments détectés dans l'image.
- Obtenez un rapport de toutes les images de votre compartiment S3 et envoyez un e-mail du rapport.

La page web appelle le service REST. Le service envoie des demandes à AWS pour effectuer les opérations suivantes :

- Obtenez et filtrez la liste des images de votre compartiment S3.
- Chargez des photos dans votre compartiment S3.
- Utilisez Amazon Rekognition pour analyser des photos individuelles et obtenir une liste d'étiquettes qui identifient les éléments détectés sur la photo.
- Analysez toutes les photos de votre compartiment S3 et utilisez Amazon SES pour envoyer un rapport par e-mail.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Détectez les personnes et les objets dans une vidéo avec Amazon Rekognition à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment détecter des personnes et des objets dans une vidéo avec Amazon Rekognition.

Java

SDK pour Java 2.x

Montre comment utiliser l'API Java Amazon Rekognition afin de créer une application qui détecte les visages et les objets dans des vidéos stockées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

JavaScript

SDK pour JavaScript (v3)

Montre comment utiliser Amazon Rekognition pour créer une application permettant de détecter AWS SDK for JavaScript des visages et des objets dans des vidéos situées dans un compartiment Amazon Simple Storage Service (Amazon S3). L'application envoie à l'administrateur une notification par e-mail contenant les résultats à l'aide d'Amazon Simple Email Service (Amazon SES).

Découvrez comment :

- Créer un utilisateur non authentifié à l'aide d'Amazon Cognito.
- Analyser les images à la recherche d'EPI à l'aide d'Amazon Rekognition.
- Vérifier une adresse e-mail pour Amazon SES.
- Envoyer une notification par e-mail à l'aide d'Amazon SES.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Amazon Rekognition
- Amazon S3
- Amazon SES

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Enregistrez les informations EXIF et autres informations sur les images à l'aide d'un SDK AWS

L'exemple de code suivant illustre comment :

- Obtenier des informations EXIF à partir d'un fichier JPG, JPEG ou PNG.
- Charger le fichier image sur un compartiment Amazon S3.
- Utiliser Amazon Rekognition pour identifier les trois principaux attributs (étiquettes) dans le fichier.
- Ajouter les informations EXIF et les étiquettes à un tableau Amazon DynamoDB dans la région.

Rust

SDK pour Rust

Obtenez les informations EXIF à partir d'un fichier JPG, JPEG ou PNG, chargez le fichier image dans un compartiment Amazon S3, utilisez Amazon Rekognition pour identifier les trois principaux attributs (étiquettes dans Amazon Rekognition) du fichier et ajoutez les informations EXIF et d'étiquettes à un tableau Amazon DynamoDB dans la région.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- DynamoDB

- Amazon Rekognition
- Amazon S3

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Transformez les données de votre application avec S3 Object Lambda

L'exemple de code suivant montre comment transformer des données pour votre application avec S3 Object Lambda.

.NET

AWS SDK for .NET

Montre comment ajouter du code personnalisé aux requêtes GET S3 standard afin de modifier l'objet demandé extrait de S3 afin que l'objet réponde aux besoins du client ou de l'application demandeur.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- Lambda
- Amazon S3

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation de ce service avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Résolution des problèmes

Cette section décrit comment résoudre les problèmes de fonctions Amazon S3 et explique comment obtenir des ID de demande dont vous avez besoin lorsque vous contactez AWS Support.

Rubriques

Résolution des erreurs d'accès refusé (403 interdit) dans Amazon S3

Important

Le 13 mai 2024, nous avons commencé à déployer une modification visant à éliminer les frais pour les demandes non autorisées qui ne sont pas initiées par le propriétaire du compartiment. Une fois le déploiement de cette modification terminé, les propriétaires de compartiments n'auront jamais à payer de frais de demande ou de bande passante pour les demandes renvoyant des erreurs `AccessDenied` (HTTP`403 Forbidden`) lorsque ces demandes sont initiées en dehors de leur AWS compte individuel ou de leur AWS organisation. Pour plus d'informations sur la liste complète des codes HTTP `3XX` et de `4XX` statut qui ne seront pas facturés, consultezFacturation des réponses aux erreurs d'Amazon S3. Cette modification de facturation ne nécessite aucune mise à jour de vos applications et s'applique à tous les compartiments S3. Lorsque le déploiement de cette modification sera complètement terminé Régions AWS, nous mettrons à jour notre documentation.

Les rubriques suivantes décrivent les causes les plus courantes des erreurs d'accès refusé (403 interdit) dans Amazon S3.

> **Note**
>
> Pour `Access Denied` (HTTP`403 Forbidden`), S3 ne facture pas le propriétaire du compartiment lorsque la demande est initiée en dehors du AWS compte individuel du propriétaire du compartiment ou de l' AWS organisation du propriétaire du compartiment.

Rubriques

> **Note**
>
> Si vous essayez de résoudre un problème d'autorisation, commencez par la section Stratégies de compartiment et politiques IAM et assurez-vous de suivre les instructions de la section Conseils pour vérifier les autorisations.

Stratégies de compartiment et politiques IAM

Opérations au niveau des compartiments

Si aucune politique de compartiment n'est en place, le compartiment autorise implicitement les demandes provenant de n'importe quelle identité AWS Identity and Access Management (IAM) du compte propriétaire du compartiment. Le compartiment refuse également implicitement les demandes

émanant de toute autre identité IAM provenant d'autres comptes, ainsi que les demandes anonymes (non signées). Toutefois, si aucune politique d'utilisateur IAM n'est en place, le demandeur (sauf s'il s'agit de l'utilisateur root) est implicitement empêché de faire des demandes. Pour plus d'informations sur cette logique d'évaluation, consultez Identification d'une demande autorisée ou refusée dans un compte dans le Guide de l'utilisateur d'IAM.

Opérations au niveau de l'objet

Si l'objet appartient au compte propriétaire du compartiment, la politique de compartiment et la politique de l'utilisateur IAM fonctionneront de la même manière pour les opérations au niveau de l'objet que pour les opérations au niveau du compartiment. Par exemple, si aucune politique de compartiment n'est en place, le compartiment autorise implicitement les demandes d'objet provenant de n'importe quelle identité IAM du compte propriétaire du compartiment. Le compartiment refuse également implicitement les demandes d'objet émanant de toute autre identité IAM provenant d'autres comptes, ainsi que les demandes anonymes (non signées). Toutefois, si aucune politique d'utilisateur IAM n'est en place, le demandeur (sauf s'il s'agit de l'utilisateur root) est implicitement empêché de faire des demandes d'objet.

Si l'objet appartient à un compte externe, l'accès à l'objet ne peut être accordé que par le biais de listes de contrôle d'accès (ACL) aux objets. La politique relative aux compartiments et la politique d'utilisateur IAM peuvent toujours être utilisées pour refuser les demandes d'objets.

Par conséquent, pour vous assurer que votre politique de compartiment ou votre politique d'utilisateur IAM n'est pas à l'origine d'une erreur d'accès refusé (403 interdit), assurez-vous que les conditions suivantes sont remplies :

- Pour l'accès au même compte, aucune déclaration `Deny` explicite ne doit être formulée à l'encontre du demandeur auquel vous essayez d'accorder des autorisations, que ce soit dans la politique de compartiment ou dans la politique d'utilisateur IAM. Si vous souhaitez accorder des autorisations en utilisant uniquement la politique de compartiment et la politique d'utilisateur IAM, l'une de ces politiques doit contenir au moins une déclaration `Allow` explicite.
- Pour l'accès intercompte, aucune déclaration `Deny` explicite ne doit être formulée à l'encontre du demandeur auquel vous essayez d'accorder des autorisations, que ce soit dans la politique de compartiment ou dans la politique d'utilisateur IAM. Si vous souhaitez accorder des autorisations intercompte en utilisant uniquement la politique de compartiment et la politique d'utilisateur IAM, la politique de compartiment et la politique d'utilisateur IAM du demandeur doivent inclure une déclaration `Allow` explicite.

> **Note**
>
> Les déclarations `Allow` d'une politique de compartiment s'appliquent uniquement aux objets appartenant au même compte propriétaire du compartiment. Toutefois, les déclarations `Deny` figurant dans une politique de compartiment s'appliquent à tous les objets, quel que soit leur propriétaire.

Pour consulter ou modifier votre politique de compartiment

> **Note**
>
> Pour afficher ou modifier une politique de compartiment, vous devez disposer de l'autorisation `s3:GetBucketPolicy`.

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment pour lequel vous souhaitez afficher ou modifier une stratégie de compartiment.
4. Choisissez l'onglet Permissions (Autorisations).
5. Sous Politique de compartiment, choisissez Modifier. La page Edit bucket policy (Modifier la politique de compartiment) s'affiche.

Pour revoir ou modifier votre politique de compartiment à l'aide de la AWS Command Line Interface (AWS CLI), utilisez la get-bucket-policycommande.

> **Note**
>
> Si l'accès à un compartiment est bloqué en raison d'une politique de compartiment incorrecte, connectez-vous au en AWS Management Console utilisant vos informations d'identification d'utilisateur root. Pour accéder de nouveau à votre compartiment, veillez à supprimer la politique de compartiment à l'aide de vos informations d'identification d'utilisateur root.

Conseils pour vérifier les autorisations

Pour vérifier si le demandeur dispose des autorisations nécessaires pour effectuer une opération Amazon S3, essayez ce qui suit :

- Identifiez le demandeur. S'il s'agit d'une demande non signée, il s'agit d'une demande anonyme sans politique d'utilisateur IAM. S'il s'agit d'une demande utilisant une URL présignée, la politique d'utilisateur sera la même que celle applicable à l'utilisateur ou au rôle IAM qui a signé la demande.
- Vérifiez que vous utilisez le bon rôle ou utilisateur IAM. Vous pouvez vérifier votre utilisateur ou rôle IAM en vérifiant le coin supérieur droit de la AWS Management Console ou à l'aide de la commande aws sts get-caller-identity.
- Vérifiez les politiques IAM attachées à l'utilisateur ou au rôle IAM. Vous pouvez choisir l'une des méthodes suivantes :
  - Testez des politiques IAM avec le simulateur de politiques IAM.
  - Passez en revue les différents types de politiques IAM.
- Si nécessaire, modifiez votre politique d'utilisateur IAM.
- Consultez les exemples suivants de politiques qui refusent ou autorisent explicitement l'accès :
  - Autoriser explicitement la politique de l'utilisateur IAM : IAM : autorise et refuse l'accès à plusieurs services par programmation et dans la console
  - Autoriser explicitement la politique de compartiment : accorder des autorisations à plusieurs comptes pour charger des objets ou définir des listes de contrôle d'accès aux objets pour l'accès public
  - Politique de refus explicite des utilisateurs IAM AWS: refuse l'accès en AWS fonction de la demande Région AWS
  - Refuser explicitement la politique de compartiment : exiger le SSE-KMS pour tous les objets enregistrés dans un compartiment

Paramètres des listes de contrôle d'accès d'Amazon S3

Lorsque vous vérifiez vos paramètres de listes ACL, vérifiez d'abord votre paramètre de propriété de l'objet pour vérifier si les listes ACL sont activées sur le compartiment. Sachez que les autorisations de listes ACL ne peuvent être utilisées que pour accorder des autorisations et ne peuvent pas être utilisées pour rejeter des demandes. Les listes ACL ne peuvent pas non plus être utilisées pour accorder l'accès à des demandeurs qui sont rejetés en raison de refus explicites dans les politiques de compartiment ou les politiques d'utilisateur IAM.

Le propriétaire du compartiment impose le paramètre de propriété de l'objet

Si le paramètre imposé par le propriétaire du compartiment est activé, il est peu probable que les paramètres des listes ACL provoquent une erreur d'accès refusé (403 interdit) car ce paramètre désactive toutes les listes ACL qui s'appliquent au compartiment et aux objets. L'application du propriétaire du compartiment est le paramètre par défaut (et recommandé) pour les compartiments Amazon S3.

Le paramètre de propriété de l'objet est défini sur le propriétaire du compartiment préféré ou le rédacteur d'objets

Les autorisations des listes ACL sont toujours valides avec le paramètre du propriétaire du compartiment préféré ou le paramètre du rédacteur d'objets. Il existe deux types de listes ACL : les listes ACL de compartiment et les listes ACL d'objets. Pour connaître les différences entre ces deux types de listes ACL, consultez Mappage des autorisations de liste ACL et de stratégie d'accès.

En fonction de l'action de la demande rejetée, vérifiez les autorisations des listes ACL pour votre compartiment ou l'objet :

- Si Amazon S3 a rejeté `LIST`, un objet `PUT`, `GetBucketAcl` ou une demande `PutBucketAcl`, passez en revue les autorisations des listes ACL pour votre compartiment.

  > **Note**
  >
  > Vous ne pouvez pas accorder d'autorisations d'objets `GET` avec les paramètres des listes ACL du compartiment.

- Si Amazon S3 a rejeté une demande `GET` concernant un objet S3 ou une demande PutObjectAcl, passez en revue les autorisations des listes ACL pour cet objet.

  > **Important**
  >
  > Si le compte qui détient l'objet est différent du compte qui détient le compartiment, l'accès à l'objet n'est pas contrôlé par la politique de compartiment.

Résolution d'une erreur d'accès refusé (403 interdit) liée à une demande d'objet **GET** lors de la propriété d'un objet intercompte

Passez en revue les paramètres de propriété de l'objet du compartiment pour déterminer le propriétaire de l'objet. Si vous avez accès aux listes ACL des objets, vous pouvez également consulter le compte du propriétaire de l'objet. (Pour consulter le compte du propriétaire de l'objet, consultez le paramètre de la liste ACL de l'objet dans la console Amazon S3.) Vous pouvez également faire une demande `GetObjectAcl` d'identification canonique du propriétaire de l'objet afin de vérifier le compte du propriétaire de l'objet. Par défaut, les listes ACL accordent des autorisations d'autorisation explicites pour les demandes `GET` adressées au compte du propriétaire de l'objet.

Après avoir confirmé que le propriétaire de l'objet est différent du propriétaire du compartiment, en fonction de votre cas d'utilisation et de votre niveau d'accès, choisissez l'une des méthodes suivantes pour résoudre l'erreur d'accès refusé (403 interdit) :

- Désactiver les listes ACL (recommandé) : cette méthode s'applique à tous les objets et peut être exécutée par le propriétaire du compartiment. Cette méthode offre automatiquement au propriétaire du compartiment la propriété de chaque objet du compartiment et leur contrôle total. Avant d'implémenter cette méthode, vérifiez les conditions préalables à la désactivation des listes ACL. Pour plus d'informations sur la façon de configurer votre compartiment en mode imposé (recommandé) par le propriétaire du compartiment, consultez Définition de la propriété d'un objet sur un compartiment existant.

  > **Important**
  >
  > Pour éviter une erreur d'accès refusé (403 interdit), veillez à migrer les autorisations de listes ACL vers une politique de compartiment avant de désactiver les listes ACL. Pour plus d'informations, consultez Bucket policy examples for migrating from ACL permissions (Exemples de politiques de compartiment pour la migration à partir d'autorisations de listes ACL).

- Remplacer le propriétaire de l'objet par le propriétaire du compartiment : cette méthode peut être appliquée à des objets individuels, mais seul le propriétaire de l'objet (ou un utilisateur disposant des autorisations appropriées) peut modifier la propriété d'un objet. Des frais PUT supplémentaires peuvent s'appliquer. (Pour plus d'informations, consultez Tarification Amazon S3.) Cette méthode confère au propriétaire du compartiment la pleine propriété de l'objet, ce qui lui permet de contrôler l'accès à l'objet par le biais d'une politique de compartiment.

Pour modifier la propriété de l'objet, effectuez l'une des opérations suivantes :

- Vous (le propriétaire du compartiment) pouvez recopier l'objet dans le compartiment.
- Vous pouvez modifier le paramètre de propriété de l'objet du compartiment en fonction du propriétaire du compartiment préféré. Si la gestion des versions est désactivée, les objets du compartiment sont remplacés. Si la gestion des versions est activée, des versions dupliquées du même objet apparaîtront dans le compartiment, et le propriétaire du compartiment peut définir une règle de cycle de vie d'expiration. Pour plus d'informations sur la modification des paramètres de propriété des objets, consultez Définition de la propriété d'un objet sur un compartiment existant.

  > **Note**
  >
  > Lorsque vous mettez à jour le paramètre de propriété de l'objet sur le propriétaire du compartiment préféré, le paramètre s'applique uniquement aux nouveaux objets chargés dans le compartiment.

- Vous pouvez demander au propriétaire de l'objet de le charger à nouveau à l'aide de la liste ACL de l'objet prédéfini `bucket-owner-full-control`.

  > **Note**
  >
  > Pour les chargements intercomptes, vous pouvez également exiger la liste ACL de l'objet prédéfini `bucket-owner-full-control` dans votre politique de compartiment. Pour un exemple de politique de compartiment, consultez Octroi d'autorisations intercomptes pour charger des objets tout en garantissant que le propriétaire du compartiment dispose d'un contrôle total.

- Conservez le rédacteur de l'objet en tant que propriétaire de l'objet : cette méthode ne modifie pas le propriétaire de l'objet, mais elle vous permet d'accorder l'accès aux objets individuellement. Pour autoriser l'accès à un objet, vous devez disposer de l'autorisation `PutObjectAcl` pour cet objet. Ensuite, pour corriger l'erreur d'accès refusé (403 interdit), ajoutez le demandeur en tant que bénéficiaire pour accéder à l'objet dans les listes ACL de l'objet. Pour plus d'informations, consultez Configuration des listes ACL.

Paramètres de blocage de l'accès public S3

Si l'échec de la demande implique un accès public ou des politiques publiques, vérifiez alors les paramètres de blocage de l'accès public S3 sur votre compte, votre compartiment ou votre point d'accès S3. À partir d'avril 2023, tous les paramètres de blocage de l'accès public sont activés par défaut pour les nouveaux compartiments. Pour plus d'informations sur comment Amazon S3 définit le terme « public », consultez La signification du mot « public ».

Lorsqu'ils sont définis sur `TRUE`, les paramètres de blocage de l'accès public agissent comme des politiques de refus explicites qui remplacent les autorisations autorisées par les listes ACL, les politiques de compartiment et les politiques d'utilisateur IAM. Pour déterminer si vos paramètres de blocage de l'accès public rejettent votre demande, examinez les scénarios suivants :

- Si la liste de contrôle d'accès (ACL) spécifiée est publique, alors le paramètre `BlockPublicAcls` rejette vos appels `PutBucketAcl` et `PutObjectACL`.
- Si la demande inclut une liste ACL publique, le paramètre `BlockPublicAcls` rejette vos appels `PutObject`.
- Si le paramètre `BlockPublicAcls` est appliqué à un compte et que la demande inclut une liste ACL publique, alors tous les appels `CreateBucket` qui incluent des listes ACL publiques échoueront.
- Si l'autorisation de votre demande est accordée uniquement par une liste ACL publique, alors le paramètre `IgnorePublicAcls` rejette la demande.
- Si la politique de compartiment spécifiée autorise l'accès public, alors le paramètre `BlockPublicPolicy` rejette vos appels `PutBucketPolicy`.
- Si le paramètre `BlockPublicPolicy` est appliqué à un point d'accès, alors tous les appels `PutAccessPointPolicy` et `PutBucketPolicy` spécifiant une politique publique et effectués via le point d'accès échoueront.
- Si le point d'accès ou le compartiment dispose d'une politique publique, le `RestrictPublicBuckets` paramètre rejette tous les appels entre comptes, à l'exception des appels Service AWS principaux. Ce paramètre rejette également tous les appels anonymes (ou non signés).

Pour consulter et mettre à jour les configurations de vos paramètres de blocage de l'accès public, consultez Configuration des paramètres de blocage d'accès public pour vos compartiments S3.

Paramètres du chiffrement Amazon S3

Amazon S3 prend en charge le chiffrement côté serveur sur votre compartiment. Le chiffrement côté serveur est le chiffrement des données à leur destination par l'application ou le service qui les reçoit. Amazon S3 chiffre vos données au niveau de l'objet lorsqu'il les écrit sur les disques des centres de AWS données et les déchiffre pour vous lorsque vous y accédez.

Par défaut, Amazon S3 applique désormais le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3) comme niveau de base du chiffrement pour chaque compartiment d'Amazon S3. Amazon S3 vous permet également de spécifier la méthode de chiffrement côté serveur lors du chargement d'objets.

Pour vérifier le statut et les paramètres de chiffrement côté serveur de votre compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le compartiment pour lequel vous souhaitez vérifier les paramètres de chiffrement.
4. Choisissez l’onglet Propriétés.
5. Faites défiler l'écran vers le bas jusqu'à la section Chiffrement par défaut, puis examinez les paramètres de Type de chiffrement.

Pour vérifier vos paramètres de chiffrement à l'aide de AWS CLI, utilisez la get-bucket-encryptioncommande.

Pour vérifier le statut du chiffrement d'un objet

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment qui contient l'objet.
4. Dans la liste Objets, choisissez le nom de l'objet pour lequel vous souhaitez ajouter ou modifier le chiffrement.

   La page de détails de l'objet s'affiche.

5. Accédez à la section Paramètres de chiffrement côté serveur pour afficher les paramètres de chiffrement côté serveur de l'objet.

Pour vérifier l'état du chiffrement de votre objet à l'aide de AWS CLI, utilisez la head-objectcommande.

Exigences en matière de chiffrement et d'autorisations

Amazon S3 prend en charge trois types de chiffrement côté serveur :

- Chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)
- Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS)
- Chiffrement côté serveur avec clés fournies par le client (SSE-C)

En fonction de vos paramètres de chiffrement, assurez-vous que les exigences d'autorisation suivantes sont remplies :

- SSE-S3 : aucune autorisation supplémentaire n'est requise.
- SSE-KMS (avec une clé gérée par le client) : pour télécharger des objets, l'autorisation `kms:GenerateDataKey` est requise sur la AWS KMS key . Pour télécharger des objets et effectuer des chargements partitionnés d'objets, l'autorisation `kms:Decrypt` est requise sur la clé KMS.
- SSE-KMS (avec un Clé gérée par AWS) — Le demandeur doit être associé au même compte que celui qui possède la `aws/s3` clé KMS. Le demandeur doit également disposer des autorisations Amazon S3 appropriées pour accéder à l'objet.
- SSE-C (avec une clé fournie par le client) : aucune autorisation supplémentaire n'est requise. Vous pouvez configurer la politique de compartiment pour exiger et restreindre le chiffrement côté serveur avec les clés de chiffrement fournies par le client pour les objets de votre compartiment.

Si l'objet est chiffré à l'aide d'une clé gérée par le client, assurez-vous que la stratégie de clé KMS vous permet d'effectuer les actions `kms:GenerateDataKey` ou `kms:Decrypt`. Pour obtenir des instructions sur la vérification de votre stratégie de clé KMS, consultez Viewing a key policy (Affichage d'une politique relative aux clés) dans le Guide du développeur AWS Key Management Service .

Paramètres de verrouillage des objets S3

Si le verrouillage des objets S3 est activé sur votre compartiment et que l'objet est protégé par une période de conservation ou une mise en suspens juridique, Amazon S3 renvoie une erreur d'accès refusé (403 interdit) lorsque vous essayez de supprimer l'objet.

Pour vérifier si le verrouillage des objets est activé sur le compartiment

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Compartiments, choisissez le nom du compartiment que vous souhaitez vérifier.
4. Choisissez l'onglet Propriétés.
5. Faites défiler jusqu'à la section Verrouillage des objets. Vérifiez si le paramètre Verrouillage des objets est Activé ou Désactivé.

Pour déterminer si l'objet est protégé par une période de conservation ou une mise en suspens juridique, consultez les informations de verrouillage de votre objet.

Si l'objet est protégé par une période de conservation ou une mise en suspens juridique, vérifiez les points suivants :

- Si la version de l'objet est protégée par le mode de conservation de la conformité, il n'est pas possible de la supprimer définitivement. Une demande `DELETE` permanente émanant de n'importe quel demandeur, y compris l'utilisateur root, se traduira par une erreur d'accès refusé (403 interdit). Sachez également que lorsque vous soumettez une demande `DELETE` pour un objet protégé par le mode de conservation de la conformité, Amazon S3 crée un marqueur de suppression pour cet objet.
- Si la version de l'objet est protégée par le mode de conservation de la gouvernance et que vous en avez l'autorisation `s3:BypassGovernanceRetention`, vous pouvez contourner la protection et supprimer définitivement la version. Pour plus d'informations, consultez Ignorer le mode de gouvernance.
- Si la version de l'objet est protégée par une mise en suspens juridique, une demande `DELETE` permanente peut entraîner une erreur d'accès refusé (403 interdit). Pour supprimer définitivement la version de l'objet, vous devez supprimer la mise en suspens juridique sur la version de l'objet. Pour supprimer une mise en suspens juridique, vous devez avoir l'autorisation

`s3:PutObjectLegalHold`. Pour plus d'informations sur la suppression d'une mise en suspens juridique, consultez Configuration du verrouillage d'objet S3.

Politique de point de terminaison d'un VPC

Si vous accédez à Amazon S3 en utilisant un point de terminaison d'un cloud privé virtuel (VPC), assurez-vous que la politique de point de terminaison du VPC ne vous empêche pas d'accéder à vos ressources Amazon S3. Par défaut, la politique de point de terminaison d'un VPC autorise toutes les demandes adressées à Amazon S3. Vous pouvez également configurer la politique de point de terminaison d'un VPC pour restreindre certaines demandes. Pour obtenir des informations sur la vérification de votre politique de point de terminaison d'un VPC, consultez Utilisation des stratégies de point de terminaison pour contrôler l'accès à des points de terminaison d'un VPC dans le Guide AWS PrivateLink .

AWS Organizations politiques

Si vous Compte AWS appartenez à une organisation, AWS Organizations les politiques peuvent vous empêcher d'accéder aux ressources Amazon S3. Par défaut, AWS Organizations les politiques ne bloquent aucune demande adressée à Amazon S3. Assurez-vous toutefois que vos AWS Organizations politiques n'ont pas été configurées pour bloquer l'accès aux compartiments S3. Pour savoir comment vérifier vos AWS Organizations politiques, consultez la section Liste de toutes les politiques dans le guide de AWS Organizations l'utilisateur.

Paramètres du point d'accès

Si vous recevez un message d'erreur d'accès refusé (403 interdit) lorsque vous effectuez des demandes via les points d'accès Amazon S3, vous devrez peut-être vérifier les points suivants :

- Les configurations de vos points d'accès
- La politique d'utilisateur IAM utilisée pour vos points d'accès
- La politique de compartiment utilisée pour gérer ou configurer vos points d'accès intercompte

Configurations et politiques des points d'accès

- Lorsque vous créez un point d'accès, vous pouvez choisir de désigner Internet ou VPC comme l'origine du réseau. Si l'origine du réseau est définie sur VPC uniquement, Amazon S3 rejettera

toutes les demandes adressées au point d'accès qui ne proviennent pas du VPC spécifié. Pour vérifier l'origine du réseau de votre point d'accès, consultez Création de points d'accès restreints à un virtual private cloud.

- Avec les points d'accès, vous pouvez également configurer des paramètres personnalisés de blocage de l'accès public, qui fonctionnent de la même manière que les paramètres de blocage de l'accès public au niveau du compartiment ou du compte. Pour vérifier vos paramètres personnalisés de blocage de l'accès public, consultez Gestion de l'accès public aux points d'accès.
- Pour envoyer des demandes réussies à Amazon S3 à l'aide de points d'accès, assurez-vous que le demandeur dispose des autorisations IAM nécessaires. Pour plus d'informations, consultez Configuration des stratégies IAM pour l'utilisation des points d'accès.
- Si la demande concerne des points d'accès intercompte, assurez-vous que le propriétaire du compartiment a mis à jour la politique du compartiment pour autoriser les demandes provenant du point d'accès. Pour plus d'informations, consultez Octroi d'autorisations pour les points d'accès intercompte.

Si l'erreur Accès refusé (403 Interdit) persiste après avoir vérifié tous les éléments de cette rubrique, récupérez votre identifiant de demande Amazon S3 et contactez-nous AWS Support pour obtenir des conseils supplémentaires.

Résolution des problèmes d'opérations par lot

Les rubriques suivantes répertorient les erreurs les plus courantes afin de vous aider à résoudre les problèmes que vous pouvez rencontrer lors d'opérations par lot.

Erreurs courantes

- Un rapport de tâche n'est pas fourni en cas de problème d'autorisation ou lorsqu'un mode de rétention du verrouillage des objets S3 est activé
- Échec de la réplication par lot S3 avec l'erreur : la génération du manifeste n'a trouvé aucune clé correspondant aux critères du filtre
- Les échecs des opérations par lot se produisent après l'ajout d'une nouvelle règle de réplication à une configuration de réplication existante
- Batch Operations échouant sur des objets avec l'erreur 400 InvalidRequest : échec de la tâche en raison d'une absence VersionId
- Créer un échec de tâche avec l'option des balises de tâche activée

- Accès refusé à la lecture du manifeste

Un rapport de tâche n'est pas fourni en cas de problème d'autorisation ou lorsqu'un mode de rétention du verrouillage des objets S3 est activé

L'erreur suivante se produit si les autorisations requises sont manquantes ou si un mode de rétention du verrouillage des objets (mode gouvernance ou mode conformité) est activé sur le compartiment de destination.

Erreur : raisons de l'échec. Le rapport de tâche n'a pas pu être écrit dans votre compartiment de rapports. Vérifiez vos autorisations.

Le rôle IAM et la politique de confiance doivent être configurés pour permettre à S3 Batch Operations d'accéder aux objets PUT dans le compartiment où le rapport sera livré. Si ces autorisations requises sont manquantes, cela entraîne un échec de livraison du rapport de tâche.

Lorsqu'un mode de rétention est activé, le bucket est protégé write-once-read-many (WORM). Le verrouillage d'objets avec un mode de rétention activé sur le compartiment de destination n'est pas pris en charge, de sorte que les tentatives de remise du rapport de fin de tâche échouent. Pour résoudre ce problème, choisissez un compartiment de destination pour vos rapports de fin de tâches pour lequel le mode de rétention du verrouillage d'objets n'est pas activé.

Échec de la réplication par lot S3 avec l'erreur : la génération du manifeste n'a trouvé aucune clé correspondant aux critères du filtre

Échec : la génération du manifeste n'a trouvé aucune clé correspondant aux critères du filtre.

Cette erreur se produit pour l'une des raisons suivantes :

- Lorsque les objets du compartiment source sont stockés dans les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive.

  Pour utiliser la réplication par lot sur ces objets, restaurez-les d'abord dans la classe de stockage S3 Standard en utilisant une opération S3 Initiate Restore Object dans une tâche d'opérations par lot. Pour plus d'informations, consultez Restauration d'un objet archivé et Restaurer des objets (opérations par lot). Une fois les objets restaurés, vous pouvez les répliquer à l'aide d'une tâche de réplication par lot.
- Lorsque les critères de filtre fournis ne correspondent à aucun objet valide dans le compartiment source.

Vérifiez et corrigez les critères de filtrage. Par exemple, dans la règle Batch Replication, le critère de filtre consiste à rechercher tous les objets du compartiment *example-s3-bucket* avec le préfixe. `Tax/` Si le nom du préfixe a été saisi de manière inexacte, avec une barre oblique au début et à la fin `/Tax/` plutôt qu'à la fin uniquement, aucun objet S3 n'a été trouvé. Pour résoudre l'erreur, corrigez le préfixe, dans ce cas, de `/Tax/` à `Tax/` dans la règle de réplication.

Les échecs des opérations par lot se produisent après l'ajout d'une nouvelle règle de réplication à une configuration de réplication existante

Les opérations par lot tentent de répliquer des objets existants pour chaque règle de la configuration de réplication du compartiment source. En cas de problème avec l'une des règles de réplication existantes, des échecs peuvent survenir.

Le rapport de fin de la tâche d'opérations par lot explique les raisons de l'échec de la tâche. Pour obtenir une liste des erreurs courantes, consultez Raisons de l'échec de la réplication Amazon S3.

Batch Operations échouant sur des objets avec l'erreur 400 InvalidRequest : échec de la tâche en raison d'une absence VersionId

L'exemple d'erreur suivant se produit si une tâche d'opérations par lot exécute des actions sur des objets d'un compartiment avec la gestion des versions et rencontre un objet dans le manifeste avec un champ d'ID de version vide.

Erreur : *BUCKET_NAME, prefix/file_name, failed,400,, La tâche a échoué car* manquante InvalidRequest VersionId

Cette erreur se produit car le champ de l'ID de version du manifeste est une chaîne vide, et non une chaîne `null` littérale.

Les opérations par lot échoueront pour cet objet ou ces objets en particulier, mais pas pour l'ensemble de la tâche. Ce problème se produit si le format du manifeste est configuré pour utiliser les ID de version pendant l'opération. Les tâches sans la gestion des versions ne rencontrent pas ce problème car elles ne fonctionnent que sur la version la plus récente de chaque objet et ignorent les ID de version figurant dans le manifeste.

Pour résoudre ce problème, convertissez les ID de version vides en chaînes `null`. Pour plus d'informations, consultez the section called “Conversion de chaînes d'ID de version vides en chaînes null”.

Créer un échec de tâche avec l'option des balises de tâche activée

Sans l'autorisation `s3:PutJobTagging`, la création de tâches d'opérations par lot avec l'option des balises de tâche activée provoque des erreurs `403 access denied`.

Pour créer des tâches Batch Operations avec l'option de balise de tâche activée, l'utilisateur AWS Identity and Access Management (IAM) qui crée la tâche Batch Operations doit disposer de l'`s3:PutJobTagging`autorisation en plus de l'`s3:CreateJob`autorisation.

Pour plus d'informations sur les autorisations requises pour les opérations par lot, consultez the section called “Octroi d'autorisations”.

Accès refusé à la lecture du manifeste

Si les opérations par lot ne parviennent pas à lire le fichier manifeste lorsque vous essayez de créer une tâche d'opérations par lot, les erreurs suivantes peuvent se produire.

AWS CLI

Motif de l'échec La lecture du manifeste est interdite : AccessDenied

Console Amazon S3

Avertissement : Impossible d'obtenir l'ETag de l'objet manifeste. Spécifiez un autre objet pour continuer.

Pour résoudre ce problème, effectuez l'une des opérations suivantes :

- Vérifiez que le rôle IAM associé à celui Compte AWS que vous avez utilisé pour créer le job Batch Operations dispose d'`s3:GetObject`autorisations. Le rôle IAM du compte doit avoir des autorisations `s3:GetObject` pour permettre aux opérations par lot de lire le fichier manifeste.

  Pour plus d'informations sur les autorisations requises pour les opérations par lot, consultez the section called “Octroi d'autorisations”.
- Vérifiez les métadonnées des objets manifestes pour détecter toute incompatibilité d'accès avec la propriété des objets S3. Pour plus d'informations sur la propriété des objets S3, consultez the section called “Contrôle de la propriété des objets”.
- Vérifiez si des clés AWS Key Management Service (AWS KMS) sont utilisées pour chiffrer le fichier manifeste.

Les opérations par lots prennent en charge les rapports d'inventaire CSV AWS KMS cryptés. Toutefois, Batch Operations ne prend pas en charge les fichiers manifestes CSV AWS KMS chiffrés. Pour plus d'informations, consultez Configuration d'Amazon S3 Inventory et Spécification d'un manifeste.

Dépannage CORS

Si vous rencontrez un comportement imprévu lorsque vous accédez aux compartiments définis avec la configuration CORS, voici les étapes à suivre :

1. Vérifiez que la configuration CORS est paramétrée sur le compartiment.

   Si la configuration CORS est définie, la console affiche un lien Edit CORS Configuration (Modifier la configuration CORS) dans la section Autorisations du compartiment Propriétés.

2. Capturez la demande et la réponse complètes grâce à l'outil de votre choix. Pour chaque demande reçue par Amazon S3, il doit exister une règle CORS correspondant aux données de la demande, comme suit :

   a. Vérifiez que la demande possède l'en-tête Origin.

      Si l'en-tête est manquant, Amazon S3 ne la traite pas comme une demande cross-origin et ne renvoie pas d'en-têtes de réponse CORS dans la réponse.

   b. Vérifiez que l'en-tête Origin dans la demande correspond à au moins l'un des éléments `AllowedOrigin` de la règle `CORSRule` spécifiée.

      Le schéma, l'hôte et les valeurs de port de l'en-tête de demande Origin doivent correspondre aux éléments `AllowedOrigin` de la règle `CORSRule`. Par exemple, si vous définissez la règle `CORSRule` pour autoriser l'origine `http://www.example.com`, les deux origines `https://www.example.com` et `http://www.example.com:80` de la demande ne correspondent pas à l'origine autorisée de votre configuration.

   c. Vérifiez que la méthode de la demande (ou, dans une demande en amont, la méthode spécifiée dans `Access-Control-Request-Method`) est l'un des éléments `AllowedMethod` de la même règle `CORSRule`.

   d. Pour une demande en amont, si la demande inclut un en-tête `Access-Control-Request-Headers`, vérifiez que la règle `CORSRule` inclut les entrées `AllowedHeader` pour chaque valeur dans l'en-tête `Access-Control-Request-Headers`.

Résolution des problèmes de cycle de vie Amazon S3

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés aux règles de cycle de vie Amazon S3.

Rubriques

J'ai exécuté une opération de liste sur mon compartiment et j'ai vu des objets qui, selon moi, avaient expiré ou avaient été transférés par une règle de cycle de vie.

Les transitions et les expirations d'objets du cycle de vie S3 sont des opérations asynchrones. Par conséquent, il peut y avoir un délai entre le moment où les objets sont éligibles à l'expiration ou à la transition et le moment où ils sont réellement transférés ou ils expirent. Les changements de tarification sont appliqués dès que la règle du cycle de vie est satisfaite, même si l'action n'est pas terminée. Il existe toutefois une exception à ce comportement si vous disposez d'une règle de cycle de vie configurée pour transférer l'objet vers la classe de stockage S3 Intelligent-Tiering. Dans ce cas, les changements de facturation ne se produisent pas tant que l'objet n'est pas passé à la classe de stockage S3 Intelligent-Tiering. Pour plus d'informations sur les changements de facturation, consultez Définition d'une configuration de cycle de vie sur un compartiment.

> **Note**
>
> Amazon S3 ne effectue pas la transition d'objets de moins de 128 Ko vers les classes de stockage S3 standard ou S3 standard – Accès peu fréquent vers les classes de stockage S3 Intelligent-Tiering, S3 standard – Accès peu fréquent ou S3 unizone – Accès peu fréquent.

Comment puis-je contrôler les mesures prises conformément à mes règles de cycle de vie ?

Pour surveiller les mesures prises par les règles du cycle de vie, vous pouvez utiliser les fonctionnalités suivantes :

- Notifications d'événements S3 — Vous pouvez configurer les notifications d'événements S3 afin d'être informé de tout événement d'expiration ou de transition du cycle de vie S3.
- Journaux d'accès au serveur S3 : vous pouvez activer les journaux d'accès au serveur pour vos compartiments S3 afin de capturer les actions du cycle de vie S3, telles que les transitions d'objets vers une autre classe de stockage ou les expirations d'objets. Pour plus d'informations, consultez Cycle de vie et journalisation.

Pour visualiser au quotidien les modifications de votre stockage causées par les actions du cycle de vie, nous vous recommandons d'utiliser les tableaux de bord S3 Storage Lens plutôt que d'utiliser CloudWatch les métriques Amazon. Dans votre tableau de bord Storage Lens, vous pouvez consulter les statistiques suivantes, qui surveillent le nombre ou la taille des objets :

- Octets de version actuelle
- Nombre d'objets de version actuelle
- Octets de version ancienne
- Nombre d'objets de version ancienne
- Nombre d'objets marqueur de suppression
- Supprimer les octets de stockage des marqueurs
- Octets de chargement partitionné non terminés
- Nombre d'objets de chargement partitionné non terminés

Le nombre de mes objets S3 continue d'augmenter, même après avoir défini des règles de cycle de vie sur un compartiment avec la gestion des versions activée.

Dans un compartiment activé pour la gestion des versions, lorsqu'un objet arrive à expiration, il n'est pas complètement supprimé du compartiment. Au lieu de cela, un marqueur de suppression est créé en tant que version la plus récente de l'objet. Les marqueurs de suppression sont toujours comptés comme des objets. Par conséquent, si une règle de cycle de vie est créée pour faire expirer uniquement les versions actuelles, alors le nombre d'objets dans le compartiment S3 augmente au lieu de diminuer.

Par exemple, supposons qu'un compartiment S3 ait la gestion des versions activée avec 100 objets et qu'une règle de cycle de vie soit définie pour faire expirer les versions actuelles de l'objet au bout de 7 jours. Après le septième jour, le nombre d'objets passe à 200 car 100 marqueurs de suppression sont créés en plus des 100 objets d'origine, qui sont désormais des versions anciennes. Pour plus d'informations sur les actions des règles de configuration du cycle de vie S3 pour les compartiments avec la gestion des versions activée, consultez Définition d'une configuration de cycle de vie sur un compartiment.

Pour supprimer définitivement des objets, ajoutez une configuration de cycle de vie supplémentaire afin de supprimer les versions précédentes des objets, les marqueurs de suppression expirés et les chargements partitionnés incomplets. Pour obtenir des instructions sur la création de nouvelles règles de cycle de vie, consultez Définition d'une configuration de cycle de vie sur un compartiment.

> **Note**
>
> - Amazon S3 arrondit la date de transition ou d'expiration d'un objet à minuit UTC le jour suivant.
>
> Lors de l'évaluation des objets pour les actions liées au cycle de vie, Amazon S3 utilise l'heure de création des objets en UTC. Prenons l'exemple d'un bucket non versionné doté d'une règle de cycle de vie configurée pour faire expirer les objets au bout d'un jour. Supposons qu'un objet ait été créé le 1er janvier à 17 h 05, heure avancée du Pacifique (PDT), ce qui correspond au 2 janvier à 00 h 05 UTC. L'objet vieillit d'un jour à 00h05 UTC le 3 janvier, ce qui le rend éligible à l'expiration lorsque S3 Lifecycle évalue les objets à 00h00 UTC le 4 janvier.

Comme les actions du cycle de vie d'Amazon S3 se produisent de manière asynchrone, il peut y avoir un certain délai entre la date spécifiée dans la règle du cycle de vie et la transition physique réelle de l'objet. Pour plus d'informations, voir Transition ou délai d'expiration.

Pour plus d'informations, consultez Règles de cycle de vie : en fonction de l'âge de l'objet.

- Pour les objets S3 protégés par le verrouillage d'objet, les versions actuelles ne sont pas supprimées définitivement. Au lieu de cela, un marqueur de suppression est ajouté aux objets, les rendant anciens. Les versions anciennes sont ensuite conservées et ne sont pas définitivement expirées.

Comment vider mon compartiment S3 en utilisant des règles de cycle de vie ?

Les règles de cycle de vie S3 constituent un outil efficace pour vider un compartiment S3 contenant des millions d'objets. Pour supprimer un grand nombre d'objets de votre compartiment S3, veillez à utiliser ces deux paires de règles de cycle de vie :

- Expirer les versions actuelles d'objets et Supprimer définitivement les versions précédentes des objets
- Supprimer les marqueurs de suppression expirés et Supprimer les téléchargements partitionnés non terminés

Pour savoir comment créer une nouvelle règle de configuration du cycle de vie, consultez Définition d'une configuration de cycle de vie sur un compartiment.

> **Note**
>
> Pour les objets S3 protégés par le verrouillage d'objet, les versions actuelles ne sont pas supprimées définitivement. Au lieu de cela, un marqueur de suppression est ajouté aux objets, les rendant anciens. Les versions anciennes sont ensuite conservées et ne sont pas définitivement expirées.

Ma facture Amazon S3 a augmenté après la transition d'objets vers une classe de stockage moins coûteuse.

Votre facture peut augmenter après le transfert d'objets vers une classe de stockage moins coûteuse pour plusieurs raisons :

- Frais généraux de S3 Glacier pour les petits objets

  Pour chaque objet vers les classes de stockage S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive Retrieval, une surcharge totale de 40 Ko est associée à cette mise à jour. Dans le cadre de la surcharge de 40 Ko, 8 Ko sont utilisés pour stocker les métadonnées et le nom de l'objet. Ces 8 Ko sont facturés selon les tarifs S3 Standard. Les 32 Ko restants sont utilisés pour l'indexation et les métadonnées associées. Ces 32 Ko sont facturés selon les tarifs S3 Glacier Flexible Retrieval ou S3 Glacier Deep Archive Retrieval.

  Par conséquent, si vous stockez de nombreux objets de petite taille, nous vous déconseillons d'utiliser des transitions de cycle de vie. Pour réduire les frais de surcharge, envisagez de regrouper de nombreux petits objets en un plus petit nombre de gros objets avant de les stocker dans Amazon S3. Pour plus d'informations sur les considérations de coûts, consultez Transition vers les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive (archivage d'objets).

- Frais de stockage minimaux

  Certaines classes de stockage S3 ont des exigences en matière de durée de stockage minimale. Les objets supprimés, remplacés ou transférés de ces classes avant que la durée minimale ne soit atteinte sont soumis à des frais de transition ou de suppression anticipés au prorata. Ces exigences en matière de durée de stockage minimale sont les suivantes :

  - S3 standard – Accès peu fréquent et S3 unizone – Accès peu fréquent : 30 jours
  - S3 Glacier Flexible Retrieval et S3 Glacier Instant Retrieval – 90 jours
  - S3 Glacier Deep Archive – 180 jours

  Pour plus d'informations sur ces exigences, consultez la section Contraintes de la Transition des objets à l'aide du cycle de vie Amazon S3. Pour plus d'informations générales sur la tarification de S3, consultez Tarification Amazon S3 et Calculateur de tarification AWS.

- Règles de transition du cycle de vie

Chaque fois qu'un objet est transféré vers une classe de stockage différente selon une règle de cycle de vie, Amazon S3 considère cette transition comme une demande de transition. Les coûts de ces demandes de transition s'ajoutent aux coûts de ces classes de stockage. Si vous envisagez de procéder à la transition d'un grand nombre d'objets, tenez compte des coûts de demande pour une transition vers un niveau inférieur. Pour plus d'informations, consultez Tarification Amazon S3.

J'ai mis à jour ma politique de compartiment, mais mes objets S3 sont toujours supprimés en raison de règles de cycle de vie expirées.

Les déclarations `Deny` figurant dans une politique de compartiment n'empêchent pas l'expiration des objets définis dans une règle de cycle de vie. Les actions du cycle de vie (telles que les transitions ou les expirations) n'utilisent pas l'opération `DeleteObject` S3. Au lieu de cela, les actions du cycle de vie S3 sont effectuées à l'aide de points de terminaison S3 internes. (Pour plus d'informations, consultez Cycle de vie et journalisation.)

Pour empêcher toute action de votre règle de cycle de vie, vous devez la modifier, la supprimer ou la désactiver.

Puis-je récupérer des objets S3 expirés conformément aux règles du cycle de vie S3 ?

Le seul moyen de récupérer des objets expirés conformément au cycle de vie S3 est de procéder à la gestion des versions, qui doit être en place avant que les objets ne puissent être éligibles à l'expiration. Vous ne pouvez pas annuler les opérations d'expiration effectuées par des règles du cycle de vie. Si des objets sont définitivement supprimés conformément aux règles du cycle de vie S3 en vigueur, vous ne pouvez pas récupérer ces objets. Pour activer la gestion des versions sur un compartiment, consultez the section called “Utilisation de la gestion des versions S3”.

Si vous avez appliqué la gestion des versions au compartiment et que les versions anciennes des objets sont toujours intactes, vous pouvez restaurer les versions précédentes des objets expirés. Pour plus d'informations sur le comportement des actions des règles de cycle de vie S3 et les états de gestion des versions, consultez le tableau Actions du cycle de vie et état du contrôle de version du compartiment dans Éléments pour décrire les actions du cycle de vie.

> **Note**
>
> Si le compartiment S3 est protégé par AWS Backup ou la réplication S3, vous pouvez également utiliser ces fonctionnalités pour récupérer vos objets expirés.

Résolution des problèmes de réplication

Cette section contient des conseils de résolution des problèmes relatifs à la réplication Amazon S3 et des informations sur les erreurs de réplication par lot S3.

Rubriques

Conseils pour la résolution des problèmes de réplication S3

Si les réplicas d'objets ne figurent pas dans le compartiment de destination après la configuration de la réplication, utilisez les conseils de dépannage suivants pour identifier les problèmes et les résoudre.

- La plupart des objets se répliquent en 15 minutes. Le temps nécessaire à Amazon S3 pour répliquer un objet dépend de plusieurs facteurs, y compris de la paire de régions source et de destination et de la taille de l'objet. Pour les objets volumineux, la réplication peut prendre plusieurs heures. Pour plus de visibilité sur les temps de réplication, vous pouvez utiliser le Contrôle du temps de réplication S3 (S3 RTC).

  Si l'objet répliqué est volumineux, attendez un moment avant de vérifier s'il apparaît dans la destination. Vous pouvez également vérifier le statut de réplication de l'objet source. Si le statut de réplication de l'objet est `PENDING`, cela signifie qu'Amazon S3 n'a pas terminé la réplication. Si le statut de réplication de l'objet est `FAILED`, vérifiez la configuration de réplication définie sur le compartiment source. En outre, pour recevoir des informations sur les échecs de réplication, vous pouvez configurer une réplication des notifications d'événements Amazon S3. Pour plus d'informations, consultez Recevoir des événements d'échec de réplication avec des notifications d'événements Amazon S3.
- Vous pouvez appeler l'opération d'API `HeadObject` pour vérifier le statut de réplication d'un objet. L'opération d'API `HeadObject` renvoie le statut de réplication d'un objet `PENDING`, `COMPLETED`

ou `FAILED`. En réponse à un appel d'API `HeadObject`, le statut de réplication est renvoyé dans l'élément `x-amz-replication-status`.

> **Note**
>
> Pour exécuter `HeadObject`, vous devez disposer d'un accès en lecture sur l'objet que vous demandez. Une demande `HEAD` possède les mêmes options qu'une demande `GET`, sans effectuer d'opération `GET`. Par exemple, pour exécuter une demande `HeadObject` à l'aide de l'AWS Command Line Interface (AWS CLI), vous pouvez exécuter la commande suivante. Remplacez *`user input placeholders`* par vos propres informations.
>
> ```
> aws s3api head-object --bucket my-bucket --key index.html
> ```

- Après que `HeadObject` renvoie les objets avec un statut de réplication `FAILED`, vous pouvez utiliser la réplication par lot S3 pour répliquer les objets ayant échoué. Vous pouvez également charger à nouveau les objets ayant échoué dans le compartiment source, ce qui lancera la réplication des nouveaux objets.
- Dans la configuration de réplication du compartiment source, procédez aux vérifications suivantes :
  - L'Amazon Resource Name (ARN) du compartiment de destination est correct.
  - Le préfixe de nom de clé est correct. A titre d'exemple, si vous définissez la configuration pour répliquer des objets avec le préfixe `Tax`, seuls les objets dotés de noms de clés `Tax/document1` ou `Tax/document2` seront répliqués. Tout objet avec le nom de clé `document3` n'est pas répliqué.
  - Le statut de la règle de réplication est `Enabled`.
- Vérifiez que la gestion des versions n'a pas été suspendue sur aucun compartiment dans la configuration de la réplication. La gestion des versions doit être activée pour les compartiments source et de destination.
- Si une règle de réplication est définie sur Remplacer la propriété de l'objet par le propriétaire du compartiment de destination, alors le rôle AWS Identity and Access Management (IAM) utilisé pour la réplication doit disposer de l'autorisation `s3:ObjectOwnerOverrideToBucketOwner`. Cette autorisation est accordée sur la ressource (dans ce cas, le compartiment de destination). Par exemple, la déclaration `Resource` suivante montre comment accorder cette autorisation sur le compartiment de destination :

```
{
  "Effect":"Allow",
```

```
  "Action":[
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource":"arn:aws:s3:::DestinationBucket/*"
}
```

- Si le compartiment de destination appartient à un autre compte, le propriétaire du compartiment de destination doit également accorder l'autorisation `s3:ObjectOwnerOverrideToBucketOwner` au propriétaire du compartiment source, conformément à la politique de compartiment de destination. Pour utiliser l'exemple de politique de compartiment suivant, remplacez *`user input placeholders`* par vos propres informations :

```
{
  "Version": "2012-10-17",
  "Id": "Policy1644945280205",
  "Statement": [
    {
      "Sid": "Stmt1644945277847",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateTags",
        "s3:ObjectOwnerOverrideToBucketOwner"
      ],
      "Resource": "arn:aws:s3:::DestinationBucket/*"
    }
  ]
}
```

> **Note**
>
> Si les paramètres de propriété de l'objet du compartiment de destination incluent Propriétaire du compartiment appliqué, vous n'avez pas besoin de mettre à jour le paramètre pour Remplacer la propriété de l'objet par le propriétaire du compartiment de destination dans la règle de réplication. Le changement de propriété de l'objet se produira par défaut. Pour plus d'informations sur la modification du propriétaire d'un réplica, consultez Modification du propriétaire d'un réplica.

- Si vous définissez la configuration de réplication dans un scénario entre comptes, dans lequel les compartiments source et de destination appartiennent à des propriétaires différentsComptes AWS, les compartiments de destination ne peuvent pas être configurés en tant que compartiments Requester Pays. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.
- Si les objets source d'un compartiment sont chiffrés à l'aide d'une clé AWS Key Management Service (AWS KMS), la règle de réplication doit être configurée pour inclure les objets chiffrés par AWS KMS. Assurez-vous de sélectionner Répliquer les objets chiffrés avec AWS KMS dans les paramètres de Chiffrement de la console Amazon S3. Sélectionnez ensuite une clé AWS KMS pour chiffrer les objets de destination.

> **Note**
>
> Si le compartiment de destination se trouve dans un autre compte, spécifiez une clé gérée par le client AWS KMS appartenant au compte de destination. N'utilisez pas la clé par défaut gérée par Amazon S3 (`aws/s3`). L'utilisation de la clé par défaut chiffre les objets à l'aide de la clé gérée par Amazon S3 appartenant au compte source, empêchant ainsi le partage de l'objet avec un autre compte. Par conséquent, le compte de destination ne pourra pas accéder aux objets du compartiment de destination.

Pour utiliser une clé AWS KMS appartenant au compte de destination afin de chiffrer les objets de destination, le compte de destination doit accorder les autorisations`kms:GenerateDataKey` et `kms:Encrypt` au rôle de réplication dans la stratégie de clé KMS. Pour utiliser l'exemple de déclaration suivant dans votre stratégie de clé KMS, remplacez *`user input placeholders`* par vos propres informations :

```
{
    "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
    },
    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
}
```

Si vous utilisez un astérisque (*) pour la déclaration `Resource` dans la stratégie de clé AWS KMS, la stratégie autorise l'utilisation de la clé KMS uniquement pour le rôle de réplication. La stratégie n'autorise pas le rôle de réplication à augmenter ses autorisations.

Par défaut, la stratégie de clé KMS accorde à l'utilisateur root des autorisations complètes sur la clé. Ces autorisations peuvent être déléguées à d'autres utilisateurs du même compte. À moins que la stratégie de clé KMS source ne contienne des déclarations `Deny`, l'utilisation d'une politique IAM pour accorder des autorisations de rôle de réplication à la clé KMS source est suffisante.

> **Note**
>
> Les stratégies de clé KMS qui limitent l'accès à des plages d'adresses CIDR, à des points de terminaison d'un VPC ou à des points d'accès S3 spécifiques peuvent entraîner l'échec de la réplication.

Si les clés KMS source ou de destination accordent des autorisations en fonction du contexte de chiffrement, vérifiez que les clés de compartiment Amazon S3 sont activées pour les compartiments. Si les clés de compartiment S3 sont activées dans les compartiments, le contexte de chiffrement doit être la ressource au niveau du compartiment, comme suit :

```
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::SOURCE_BUCKET_NAME"
     ]
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::DESTINATION_BUCKET_NAME"
     ]
```

Outre les autorisations accordées par la stratégie de clé KMS, le compte source doit ajouter les autorisations minimales suivantes à la politique IAM du rôle de réplication :

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": [
        "SourceKmsKeyArn"
```

```
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Encrypt"
    ],
    "Resource": [
        "DestinationKmsKeyArn"
    ]
}
```

Pour plus d'informations sur la réplication des objets chiffrés avec AWS KMS, consultez Réplication d'objets chiffrés.

- Si le compartiment de destination appartient à un autre Compte AWS, vérifiez si le propriétaire du compartiment dispose d'une stratégie de compartiment sur le compartiment de destination, qui permet au propriétaire du compartiment source de répliquer des objets. Pour obtenir un exemple, consultez Configuration d'une réplication quand les compartiments source et de destination appartiennent à des comptes distincts.
- Si vos objets ne se répliquent toujours pas une fois que vous avez validé les autorisations, vérifiez la présence de déclarations `Deny` explicites dans les emplacements suivants :
  - Les déclarations `Deny` figurant dans les politiques du compartiment source ou de destination. La réplication échoue si la stratégie de compartiment refuse l'accès au rôle de réplication pour l'une des actions suivantes :

    Compartiment source :

    ```
            "s3:GetReplicationConfiguration",
            "s3:ListBucket",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
    ```

    Compartiments de destination :

```
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
```

  - Les déclarations `Deny` ou les limites des autorisations associées au rôle IAM peuvent entraîner l'échec de la réplication.
  - Les déclarations `Deny` figurant dans les politiques de contrôle des services AWS Organizations associées aux comptes source ou de destination peuvent entraîner l'échec de la réplication.
- Si un réplica d'objet ne figure pas dans le compartiment de destination, les problèmes suivants ont pu empêcher la réplication :
  - Amazon S3 ne réplique pas un objet figurant dans un compartiment source qui est lui-même un réplica créé par une autre configuration de réplication. Par exemple, si vous définissez une configuration de réplication à partir du compartiment A vers le compartiment B vers le compartiment C, Amazon S3 ne réplique pas les réplicas d'objets dans le compartiment B vers le compartiment C.
  - Un propriétaire de compartiment source peut accorder à d'autres Comptes AWS l'autorisation de charger des objets. Par défaut, le propriétaire du compartiment source ne possède aucune autorisation pour les objets créés par d'autres comptes. La configuration de réplication réplique uniquement les objets pour lesquels le propriétaire du compartiment source dispose des autorisations d'accès. Le propriétaire du compartiment source peut accorder à d'autres Comptes AWS les autorisations permettant de créer des objets de manière conditionnelle, en exigeant des autorisations d'accès explicites sur ces objets. Pour un exemple de politique, consultez Octroi d'autorisations intercomptes pour charger des objets tout en garantissant que le propriétaire du compartiment dispose d'un contrôle total.
- Supposons que vous ajoutiez une règle dans la configuration de réplication pour répliquer un sous-ensemble d'objets dotés d'une balise spécifique. Dans ce cas, vous devez attribuer une clé et une valeur de balise spécifiques au moment de la création de l'objet pour qu'Amazon S3 puisse répliquer l'objet. Si vous commencez par créer un objet, puis ajoutez la balise à l'objet existant, Amazon S3 ne réplique pas l'objet.
- Les notifications d'événement Amazon S3 pour vous avertir dans les cas où les objets ne sont pas répliqués vers leur Région AWS de destination. Les notifications d'événements Amazon S3 sont disponibles via Amazon Simple Queue Service (Amazon SQS), Amazon Simple Notification Service (Amazon SNS) ou AWS Lambda. Pour plus d'informations, consultez Recevoir des événements d'échec de réplication avec des notifications d'événements Amazon S3.

Vous pouvez également consulter les raisons de l'échec de la réplication en utilisant les notifications d'événements Amazon S3. Pour consulter la liste des raisons d'échec, consultez Raisons de l'échec de la réplication Amazon S3.

Erreurs de réplication par lot

Pour résoudre les problèmes liés aux objets qui ne se répliquent pas vers le compartiment de destination, vérifiez les différents types d'autorisations pour le compartiment, le rôle de réplication et le rôle IAM utilisés pour créer la tâche de réplication par lot. Assurez-vous également de vérifier les paramètres d'accès public et les paramètres de propriété des compartiments.

Vous pouvez rencontrer l'une des erreurs suivantes avec la réplication par lot :

- L'opération par lot a un statut d'échec pour une raison : le rapport de tâche n'a pas pu être écrit dans votre compartiment de rapports.

  Cette erreur se produit si le rôle IAM utilisé pour la tâche d'opérations par lot ne parvient pas à placer le rapport de fin à l'emplacement spécifié lors de la création de la tâche. Pour résoudre cette erreur, vérifiez que le rôle IAM dispose des autorisations `PutObject` nécessaires pour le compartiment dans lequel vous souhaitez enregistrer le rapport de fin des opérations par lot. C'est une bonne pratique que de transmettre le rapport dans un compartiment autre que le compartiment source.
- L'opération par lot est terminée avec des échecs et le nombre total d'échecs n'est pas égal à 0.

  Cette erreur se produit en cas de problèmes d'autorisations d'objet insuffisantes avec la tâche de réplication par lot en cours d'exécution. Si vous utilisez une règle de réplication pour votre tâche de réplication par lot, assurez-vous que le rôle IAM utilisé pour la réplication dispose des autorisations appropriées pour accéder aux objets depuis le compartiment source ou de destination. Vous pouvez également consulter le Rapport de fin de la réplication par lot pour connaître les raisons spécifiques de l'échec de la réplication Amazon S3.
- La tâche par lot s'est correctement exécutée, mais le nombre d'objets attendus dans le compartiment de destination n'est pas le même.

  Cette erreur se produit lorsqu'il existe une incompatibilité entre les objets répertoriés dans le manifeste fourni dans la tâche de réplication par lot et les filtres que vous avez sélectionnés lors de la création de la tâche. Vous pouvez également recevoir ce message lorsque les objets de votre

compartiment source ne correspondent à aucune règle de réplication et ne sont pas inclus dans le manifeste généré.

Résolution des problèmes de journalisation des accès au serveur

Les rubriques suivantes peuvent vous aider à résoudre les problèmes que vous êtes susceptible de rencontrer lors de la configuration de la journalisation avec Amazon S3.

Rubriques

Messages d'erreur courants lors de la configuration de la journalisation

Les messages d'erreur courants suivants peuvent s'afficher lorsque vous activez la journalisation via l'AWS Command Line Interface (AWS CLI) et les kits SDK AWS :

Erreur : la journalisation entre localisations S3 n'est pas autorisée

Si le compartiment de destination (également appelé compartiment cible) se trouve dans une région différente de celle du compartiment source, une erreur La journalisation entre localisations S3 n’est pas autorisée se produit. Pour résoudre cette erreur, assurez-vous que le compartiment de destination configuré pour recevoir les journaux d’accès se trouve dans les mêmes Région AWS et Compte AWS que le compartiment source.

Erreur : le propriétaire du compartiment à journaliser et celui du compartiment cible doivent être identiques

Lorsque vous activez la journalisation des accès au serveur, cette erreur se produit si le compartiment de destination spécifié appartient à un compte différent. Pour résoudre cette erreur, assurez-vous que le compartiment de destination se trouve dans le même Compte AWS que le compartiment source.

> **Note**
>
> Nous vous recommandons de choisir un compartiment de destination différent du compartiment source. Lorsque le compartiment source et le compartiment de destination sont identiques, des journaux supplémentaires sont créés pour les journaux qui sont écrits

dans le compartiment, ce qui peut augmenter votre facture de stockage. Ces journaux supplémentaires peuvent également rendre difficile de trouver des journaux spécifiques. Pour simplifier la gestion des journaux, nous vous recommandons d'enregistrer les journaux d'accès dans un autre compartiment. Pour plus d'informations, consultez the section called "Comment activer la livraison des journaux ?".

Erreur : le compartiment cible pour la journalisation n'existe pas

Le compartiment de destination doit exister avant de définir la configuration. Cette erreur indique que le compartiment de destination n'existe pas ou est introuvable. Assurez-vous que le nom du compartiment est correctement orthographié, puis réessayez.

Erreur : les accords cibles ne sont pas autorisés pour les compartiments appliqués par le propriétaire du compartiment

Cette erreur indique que le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour la propriété des objets S3. Le paramètre Propriétaire du compartiment appliqué ne prend pas en charge les octrois de destination (cibles). Pour plus d'informations, consultez Autorisations de diffusion de journaux.

Dépannage des échecs de livraison

Pour éviter les problèmes de journalisation des accès au serveur, veillez à suivre les bonnes pratiques suivantes :

- Le groupe de livraison des journaux S3 dispose d'un accès en écriture au compartiment de destination : le groupe de livraison des journaux S3 fournit des journaux d'accès au serveur vers le compartiment de destination. Une politique de compartiment ou une liste de contrôle d'accès (ACL) peuvent être utilisées pour accorder l'accès en écriture au compartiment de destination. Toutefois, nous vous recommandons d'utiliser une stratégie de compartiment plutôt qu'une liste ACL. Pour plus d'informations sur la manière d'accorder l'accès en écriture à votre compartiment de destination, consultez Autorisations de diffusion de journaux.

  > **Note**
  >
  > Si le compartiment de destination utilise le paramètre Propriétaire du compartiment appliqué pour Propriété d'objets, tenez compte de ce qui suit :

- Les listes ACL sont désactivées et n'affectent plus les autorisations. Vous ne pouvez pas mettre à jour la liste ACL de votre compartiment pour accorder l'accès au groupe de livraison des journaux S3. À la place, pour accorder l’accès au principal du service de journalisation, vous devez mettre à jour la politique de compartiment pour le compartiment de destination.
- Vous ne pouvez pas inclure d’octrois de destination dans votre configuration `PutBucketLogging`.

- La politique de compartiment du compartiment de destination permet d’accéder aux journaux : vérifiez la politique de compartiment du compartiment de destination. Recherchez dans la politique du compartiment toutes les déclarations contenant `"Effect": "Deny"`. Vérifiez ensuite que la déclaration `Deny` n'empêche pas l'écriture des journaux d'accès dans le compartiment.
- Le verrouillage d’objet S3 n’est pas activé sur le compartiment de destination : vérifiez si le verrouillage d’objet est activé dans le compartiment de destination. Le verrouillage des objets bloque la livraison des journaux d'accès au serveur. Vous devez choisir un compartiment de destination sur lequel le verrouillage d’objet n’est pas activé.
- Les clés gérées par Amazon S3 (SSE-S3) sont sélectionnées si le chiffrement par défaut est activé sur le compartiment de destination : vous pouvez utiliser le chiffrement de compartiment par défaut sur le compartiment de destination uniquement si vous utilisez le chiffrement côté serveur avec les clés gérées par Amazon S3 (SSE-S3). Le chiffrement côté serveur par défaut avec les clés AWS Key Management Service (AWS KMS) (SSE-KMS) n’est pas pris en charge pour les compartiments de destination de journalisation des accès au serveur. Pour plus d'informations sur l'activation du chiffrement par défaut, consultez Configuration du chiffrement par défaut.
- Le paiement par le demandeur n’est pas activé dans le compartiment de destination : le paiement par le demandeur n’est pas pris en charge dans le compartiment de destination pour la journalisation des accès au serveur. Pour autoriser la livraison des journaux d’accès au serveur, désactivez l’option Paiement par le demandeur sur le compartiment de destination.
- Passez en revue votre politique de contrôle du service AWS Organizations : lorsque vous utilisez AWS Organizations, vérifiez les politiques de contrôle des services pour vous assurer que l'accès à Amazon S3 est autorisé. Les politiques de contrôle des services spécifient les autorisations maximales pour les comptes concernés. Recherchez toutes les déclarations qui contiennent `"Effect": "Deny"` dans la politique de contrôle des services et vérifiez que les déclarations `Deny` n'empêchent pas l'écriture de journaux d'accès dans le compartiment. Pour plus d'informations, consultez Politiques de contrôle de service (SCP) dans le Guide de l'utilisateur AWS Organizations.

- Prévoyez un certain temps pour que les modifications récentes de la configuration de la journalisation prennent effet : l'activation de la journalisation des accès au serveur pour la première fois ou la modification du compartiment de destination pour les journaux nécessite du temps pour entrer pleinement en vigueur. Plus d'une heure peut être nécessaire pour que toutes les demandes soient correctement journalisées et livrées.

  Pour vérifier les échecs de livraison des journaux, activez les métriques de demande sur Amazon CloudWatch. Si les journaux ne sont pas livrés au bout de quelques heures, recherchez la métrique `4xxErrors`, qui peut indiquer des échecs de livraison des journaux. Pour plus d'informations sur l'activation des métriques de demande, consultez the section called "Création d'une configuration de métriques pour tous les objets".

Résolution des problèmes de gestion des versions

Les rubriques suivantes peuvent vous aider à résoudre des problèmes courants de gestion des versions sur Amazon S3.

Rubriques

Je souhaite récupérer des objets qui ont été supprimés accidentellement dans un compartiment avec la gestion des versions.

En général, lorsque des versions d'objets sont supprimées des compartiments S3, Amazon S3 n'a aucun moyen de les récupérer. Toutefois, si vous avez activé lea gestion des versions S3 sur votre compartiment S3, une demande `DELETE` qui ne spécifie pas d'ID de version ne peut pas supprimer définitivement un objet. Au lieu de cela, un marqueur de suppression est ajouté en tant qu'espace réservé. Ce marqueur de suppression devient la version actuelle de l'objet.

Pour vérifier si vos objets supprimés sont définitivement ou temporairement supprimés (avec un marqueur de suppression à leur place), procédez comme suit :

1. Connectez-vous à la console Amazon S3 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/s3/.
2. Dans le panneau de navigation de gauche, choisissez Compartiments.
3. Dans la liste Buckets (Compartiments), choisissez le nom du compartiment qui contient l'objet.
4. Dans la liste Objets, activez le bouton Afficher les versions à droite de la barre de recherche, puis recherchez l'objet supprimé dans la barre de recherche. Cette option n'est disponible que si la gestion des versions a précédemment été activée sur le compartiment.

   Vous pouvez également utiliser S3 Inventory pour rechercher des objets supprimés.
5. Si vous ne trouvez pas l'objet après avoir activé l'option Afficher les versions ou créé un rapport d'inventaire, et que vous ne trouvez pas non plus de marqueur de suppression de l'objet, la suppression est définitive et l'objet ne peut pas être récupéré.

Vous pouvez également vérifier le statut d'un objet supprimé à l'aide de l'opération `HeadObject` API depuis le AWS Command Line Interface (AWS CLI). Pour ce faire, utilisez la commande `head-object` suivante et remplacez *`user input placeholders`* par vos propres informations :

```
aws s3api head-object --bucket example-s3-bucket --key index.html
```

Si vous exécutez la commande `head-object` sur un objet avec la gestion des versions dont la version actuelle est un marqueur de suppression, vous recevrez une erreur 404 Introuvable. Par exemple :

Une erreur s'est produite (404) lors de l'appel de l' HeadObject opération : Introuvable

Si vous exécutez la commande `head-object` sur un objet avec la gestion des versions et que vous fournissez l'ID de version de l'objet, Amazon S3 récupère les métadonnées de l'objet, confirmant ainsi que l'objet existe toujours et qu'il n'est pas supprimé définitivement.

```
aws s3api head-object --bucket example-s3-bucket --key index.html --
version-id versionID
```

```
{
"AcceptRanges": "bytes",
"ContentType": "text/html",
"LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
"ContentLength": 77,
"VersionId": "Zg5HyL7m.eZU9iM7AVlJkrqAiE.0UG4q",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
```

```
"Metadata": {}
}
```

Si l'objet est trouvé et que la version la plus récente est un marqueur de suppression, la version précédente de l'objet existe toujours. Le marqueur de suppression étant la version actuelle de l'objet, vous pouvez récupérer l'objet en supprimant le marqueur de suppression.

Une fois que vous avez supprimé définitivement le marqueur de suppression, la deuxième version la plus récente de l'objet devient la version actuelle de l'objet, ce qui rend votre objet à nouveau disponible. Pour une représentation visuelle de la façon dont les objets sont récupérés, consultez Suppression des marqueurs de suppression.

Pour supprimer une version spécifique d'un objet, vous devez être le propriétaire du compartiment. Pour supprimer un marqueur de suppression définitivement, vous devez inclure son ID de version dans une demande `DeleteObject`. Pour supprimer le marqueur de suppression, utilisez la commande suivante et remplacez *`user input placeholders`* par vos propres informations :

```
aws s3api delete-object --bucket example-s3-bucket --key index.html --
version-id versionID
```

Pour plus d'informations sur la commande `delete-object`, consultez delete-object dans la Référence de commande de l'AWS CLI . Pour plus d'informations sur la suppression permanente de marqueurs de suppression, consultez Gestion des marqueurs de suppression.

Je souhaite supprimer définitivement les objets avec la gestion des versions

Dans un compartiment avec la gestion des versions, une demande `DELETE` sans ID de version ne peut pas supprimer de façon permanente un objet. Au lieu de cela, une telle demande insère un marqueur de suppression.

Pour supprimer définitivement des objets avec la gestion des versions, vous pouvez choisir l'une des méthodes suivantes :

- Créez une règle de cycle de vie S3 pour supprimer définitivement les anciennes versions. Pour supprimer définitivement les versions anciennes d'objets, sous Supprimer définitivement les anciennes versions des objets, dans Jours après lesquels les objets deviennent anciens, saisissez le nombre de jours. Vous pouvez éventuellement spécifier le nombre de versions plus récentes à conserver en saisissant une valeur sous Number of newer versions to retain (Nombre de versions plus récentes à conserver). Pour plus d'informations sur la création de cette règle, consultez Setting an S3 Lifecycle configuration (Définition d'une configuration du cycle de vie S3).

- Supprimez une version spécifiée en incluant l'ID de version dans la demande DELETE. Pour plus d'informations, consultez How to delete versioned objects permanently (Comment supprimer des objets avec la gestion des versions de façon permanente).
- Créez une règle de cycle de vie pour faire expirer les versions actuelles. Pour faire expirer les versions actuelles des objets, sous Expirer les versions actuelles d'objets, dans Jours après la création de l'objet, saisissez le nombre de jours. Pour plus d'informations sur la création de cette règle de cycle de vie, consultez Setting an S3 Lifecycle configuration (Définition d'une configuration du cycle de vie S3).
- Pour supprimer définitivement tous les objets avec la gestion des versions et supprimer les marqueurs, créez deux règles de cycle de vie : l'une pour faire expirer les versions actuelles et supprimer définitivement les versions anciennes des objets, et l'autre pour supprimer les marqueurs de suppression d'objets expirés.

Dans un compartiment avec la gestion des versions, une demande DELETE qui ne spécifie pas d'ID de version ne peut supprimer que les objets dotés d'un identifiant de version NULL. Si l'objet a été chargé lorsque la gestion des versions était activée, une demande DELETE qui ne spécifie pas d'ID de version crée un marqueur de suppression de cet objet.

> **Note**
>
> Pour les compartiments avec le verrouillage des objets S3, une demande d'objet DELETE avec un ID de version d'objet protégé provoque une erreur 403 Accès refusé. Une demande d'objet DELETE sans ID de version ajoute un marqueur de suppression en tant que version la plus récente de l'objet avec une réponse 200 OK. Les objets protégés par le verrouillage des objets ne peuvent pas être supprimés définitivement tant que leurs périodes de conservation et leurs mises en suspens juridiques ne sont pas levées. Pour plus d'informations, consultez the section called “Fonctionnement du verrouillage d'objets S3”.

Je constate une dégradation des performances après avoir activé la gestion des versions sur les compartiments

Une dégradation des performances peut se produire sur les compartiments avec la gestion des versions s'il y a trop de marqueurs de suppression ou d'objets avec la gestion des versions et si les meilleures pratiques ne sont pas suivies.

Marqueurs de suppression trop nombreux

Après avoir activé la gestion des versions sur un compartiment, une demande DELETE faite à un objet sans ID de version crée un marqueur de suppression avec un ID de version unique. Les configurations de cycle de vie avec une règle Expirer les versions actuelles d'objets ajoutent un marqueur de suppression avec un ID de version unique à chaque objet. Un nombre excessif de marqueurs de suppression peut réduire les performances du compartiment.

Lorsque la gestion des versions est suspendue sur un compartiment, Amazon S3 marque l'ID de version comme NULL sur les nouveaux objets créés. Dans un compartiment avec la gestion des versions suspendue, l'action d'expiration entraîne la création par Amazon S3 d'un marqueur de suppression avec l'ID de version NULL. Dans un compartiment avec la gestion des versions suspendue, un marqueur de suppression NULL est créé pour chaque demande de suppression. Ces marqueurs de suppression NULL sont également appelés marqueurs de suppression d'objet expiré lorsque toutes les versions d'objet sont supprimées et qu'il ne reste qu'un seul marqueur de suppression. Si trop de marqueurs de suppression NULL s'accumulent, les performances du compartiment se dégradent.

Objets avec la gestion des versions trop nombreux

Si un compartiment avec la gestion des versions contient des objets contenant des millions de versions, le nombre d'erreurs 503 Service indisponible peut augmenter. Si vous remarquez une augmentation importante du nombre de réponses HTTP 503 Service indisponible reçues pour des demandes d'objet PUT ou DELETE vers un compartiment avec la gestion des versions, il est possible qu'un ou plusieurs objets du compartiment aient des millions de versions. Lorsque vous avez des objets avec des millions de versions, Amazon S3 limite automatiquement les demandes vers le compartiment. La limitation des demandes protège votre compartiment d'une quantité excessive de trafic de demandes qui pourrait potentiellement gêner d'autres demandes effectuées auprès du même compartiment.

Pour déterminer quels objets ont des millions de versions, utilisez S3 Inventory. S3 Inventory génère un rapport qui fournit une liste de fichiers plats des objets d'un compartiment. Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).

Pour vérifier si le compartiment contient un nombre élevé d'objets avec la gestion des versions, utilisez les métriques de S3 Storage Lens pour afficher Nombre d'objets de version actuelle Nombre d'objets de version ancienne et Nombre d'objets marqueur de suppression. Pour plus d'informations sur les métriques de Storage Lens, consultez Glossaire des métriques Amazon S3 Storage Lens.

L'équipe Amazon S3 encourage les clients à examiner les applications qui remplacent souvent le même objet et créent potentiellement des millions de versions de cet objet, afin de déterminer si

l'application fonctionne comme prévu. Par exemple, une application qui remplace le même objet toutes les minutes pendant une semaine peut créer plus de dix mille versions. Nous recommandons de stocker moins de cent mille versions pour chaque objet. Si votre cas d'utilisation nécessite des millions de versions pour un ou plusieurs objets, contactez l' AWS Support équipe pour obtenir de l'aide afin de déterminer la meilleure solution.

Bonnes pratiques

Pour éviter les problèmes de dégradation des performances liés à la gestion des versions, nous vous recommandons de suivre les bonnes pratiques suivantes :

- Activez une règle de cycle de vie pour faire expirer les anciennes versions des objets. Par exemple, vous pouvez créer une règle de cycle de vie pour faire expirer les versions anciennes 30 jours après la fin de la période d'inactivité de l'objet. Vous pouvez également conserver plusieurs anciennes versions si vous ne souhaitez pas toutes les supprimer. Pour plus d'informations, consultez Setting an S3 Lifecycle configuration (Définition d'une configuration du cycle de vie S3).
- Activez une règle de cycle de vie pour supprimer les marqueurs de suppression d'objets expirés auxquels aucun objet de données n'est associé dans le compartiment. Pour plus d'informations, consultez Suppression des marqueurs de suppression d'objet expiré.

Pour en savoir plus sur les bonnes pratiques d'optimisation des performances d'Amazon S3, consultez Schémas de conception des bonnes pratiques.

Obtenir les identifiants de demande Amazon S3 pour AWS Support

Chaque fois que vous contactez AWS Support parce que vous avez rencontré des erreurs ou un comportement inattendu dans Amazon S3, vous devez fournir les identifiants de demande associés à l'action qui a échoué. AWS Support utilise ces identifiants de demande pour résoudre les problèmes que vous rencontrez.

Les ID de demande viennent par paires, sont renvoyés dans chaque réponse traitée par Amazon S3 (même ceux contenant des erreurs), et sont accessibles via des journaux verbeux. Il existe un certain nombre de méthodes courantes pour obtenir vos identifiants de demande, notamment les journaux d'accès et les AWS CloudTrail événements S3 ou les événements de données.

Après avoir récupéré ces journaux, copiez et conservez ces deux valeurs, car vous en aurez besoin lors de votre contact AWS Support. Pour plus d'informations sur la prise de contact AWS Support, consultez la section Contact AWS ou la AWS Support documentation.

Utilisation de HTTP pour obtenir des ID de demande

Vous pouvez obtenir les ID de demande, `x-amz-request-id` et `x-amz-id-2` en consignant les bits d'une demande HTTP avant qu'elle n'atteigne l'application cible. Plusieurs outils tiers peuvent être utilisés pour récupérer les journaux verbeux pour les demandes HTTP. Choisissez un outil fiable, exécutez-le pour écouter le port sur lequel le trafic Amazon S3 circule, lorsque vous envoyez une autre demande HTTP Amazon S3.

Pour les demandes HTTP, la paire d'ID de demande ressemble à ce qui suit :

```
x-amz-request-id: 79104EXAMPLEB723
x-amz-id-2: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km
```

> **Note**
>
> Les demandes HTTPS sont chiffrées et masquées dans la plupart des captures de paquet.

Utilisation d'un navigateur web pour obtenir des ID de demande

La plupart des navigateurs web possèdent des outils de développement que vous pouvez utiliser pour consulter les en-têtes de demande.

Pour les demandes basées sur navigateur Web qui renvoient une erreur, la paire d'ID de demande ressemble aux exemples suivants.

```
<Error><Code>AccessDenied</Code><Message>Access Denied</Message>
<RequestId>79104EXAMPLEB723</RequestId><HostId>IOWQ4fDEXAMPLEQM
+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK+Jd1vEXAMPLEa3Km</HostId></Error>
```

Pour obtenir la paire d'ID de demande à partir de demandes réussies, utilisez les outils de développeur de votre navigateur pour consulter les en-têtes de réponse HTTP. Pour plus d'informations sur les outils de développeur pour les navigateurs spécifiques, consultez Amazon S3 Troubleshooting - How to recover your S3 request IDs (Dépannage Amazon S3 : récupération des valeurs d'ID de demande S3) dans AWS re:Post.

Utilisation des AWS SDK pour obtenir les identifiants de demande

Les sections suivantes incluent des informations pour configurer la journalisation grâce à un kit SDK AWS . Bien que vous puissiez activer la journalisation verbeuse sur chaque demande et réponse, nous ne vous recommandons pas d'activer la journalisation dans les systèmes de production car les demandes ou réponses volumineuses peuvent entraîner le ralentissement d'une application.

Pour les demandes du AWS SDK, la paire d'identifiants de demande ressemblera aux exemples suivants.

```
Status Code: 403, AWS Service: Amazon S3, AWS Request ID: 79104EXAMPLEB723
AWS Error Code: AccessDenied  AWS Error Message: Access Denied
S3 Extended Request ID: IOWQ4fDEXAMPLEQM+ey7N9WgVhSnQ6JEXAMPLEZb7hSQDASK
+Jd1vEXAMPLEa3Km
```

Utilisation du SDK for Go pour obtenir les identifiants de demande

Vous pouvez configurer la journalisation à l'aide du SDK for Go. Pour plus d'informations, consultez la section Métadonnées des réponses dans le guide du développeur du SDK for Go V2.

Utilisation du kit SDK pour PHP pour obtenir des ID de demande

Vous pouvez configurer la journalisation grâce à PHP. Pour plus d'informations, consultez How can I see what data is sent over the wire? (Comment savoir quelles données sont envoyées sur le réseau ?) dans le Guide du développeur AWS SDK for PHP .

Utilisation du kit SDK pour Java pour obtenir des ID de demande

Vous pouvez activer la journalisation pour des demandes ou réponses spécifiques afin de récupérer et de renvoyer uniquement les en-têtes pertinents. Pour ce faire, importez la classe `com.amazonaws.services.s3.S3ResponseMetadata`. Ensuite, vous pouvez stocker la demande dans une variable avant de faire la demande réelle. Appelez `getCachedResponseMetadata(AmazonWebServiceRequest request).getRequestID()` pour obtenir la demande ou réponse journalisée.

Example

```
PutObjectRequest req = new PutObjectRequest(bucketName, key, createSampleFile());
s3.putObject(req);
```

```
S3ResponseMetadata md = s3.getCachedResponseMetadata(req);
System.out.println("Host ID: " + md.getHostId() + " RequestID: " + md.getRequestId());
```

Sinon, vous pouvez utiliser une journalisation verbeuse de chaque demande ou réponse Java. Pour plus d'informations, consultez Verbose Wire Logging (Journalisation du réseau filaire détaillée) dans le Guide du développeur AWS SDK for Java .

Utilisation du AWS SDK for .NET pour obtenir les identifiants de demande

Vous pouvez configurer la journalisation avec le AWS SDK for .NET en utilisant l'outil de `System.Diagnostics` journalisation intégré. Pour plus d'informations, consultez le billet de blog consacré à la journalisation avec le AWS SDK pour les développeurs AWS .NET.

> **Note**
>
> Par défaut, le journal renvoyé contient uniquement des informations sur les erreurs. Pour obtenir les ID de demande, le fichier de configuration doit avoir l'élément `AWSLogMetrics` (et si vous le souhaitez, `AWSResponseLogging`) ajouté.

Utilisation du kit SDK pour Python (Boto3) pour obtenir des ID de demande

Avec le AWS SDK for Python (Boto3), vous pouvez enregistrer des réponses spécifiques. Vous pouvez utiliser cette fonction pour capturer uniquement les en-têtes pertinents. Le code suivant vous montre comment journaliser des parties de la réponse à un fichier :

```
import logging
import boto3
logging.basicConfig(filename='logfile.txt', level=logging.INFO)
logger = logging.getLogger(__name__)
s3 = boto3.resource('s3')
response = s3.Bucket(bucket_name).Object(object_key).put()
logger.info("HTTPStatusCode: %s", response['ResponseMetadata']['HTTPStatusCode'])
logger.info("RequestId: %s", response['ResponseMetadata']['RequestId'])
logger.info("HostId: %s", response['ResponseMetadata']['HostId'])
logger.info("Date: %s", response['ResponseMetadata']['HTTPHeaders']['date'])
```

Vous pouvez également intercepter les exceptions et consigner les informations pertinentes lorsqu'une exception est déclenchée. Pour plus d'informations, consultez Discerning useful

information from error responses (Obtenir des informations utiles à partir d'erreurs de réponse) dans la Référence d'API du kit SDK AWS for Python (Boto).

En outre, vous pouvez configurer Boto3 pour générer des journaux de débogage verbeux à l'aide du code suivant :

```
import boto3
boto3.set_stream_logger('', logging.DEBUG)
```

Pour plus d'informations, consultez set_stream_logger dans la Référence d'API du kit SDK AWS for Python (Boto).

Utilisation du kit SDK for Ruby pour obtenir des ID de demande

Vous pouvez obtenir vos ID de demande grâce au kit SDK for Ruby versions 1, 2 ou 3.

- Utilisation du kit SDK pour Ruby - Version 1 : vous pouvez activer la journalisation du réseau filaire HTTP à l'échelle internationale avec la ligne de code suivante.

  ```
  s3 = AWS::S3.new(:logger => Logger.new($stdout), :http_wire_trace => true)
  ```

- Utilisation du kit SDK pour Ruby - Version 2 ou Version 3 : vous pouvez activer la journalisation du réseau filaire HTTP à l'échelle internationale avec la ligne de code suivante.

  ```
  s3 = Aws::S3::Client.new(:logger => Logger.new($stdout), :http_wire_trace => true)
  ```

Pour obtenir des conseils sur l'obtention d'informations de virement auprès d'un AWS client, voir Conseil de débogage : obtenir des informations de traçage de câbles auprès d'un client.

Utilisation du AWS CLI pour obtenir les identifiants de demande

Pour obtenir vos identifiants de demande lorsque vous utilisez le AWS Command Line Interface (AWS CLI), ajoutez-le `--debug` à votre commande.

Utilisation de Windows PowerShell pour obtenir les ID de demande

Pour plus d'informations sur la restauration des journaux avec Windows PowerShell, consultez le billet de blog Response Logging in AWS Tools for Windows PowerShell .NET Development.

Utilisation d'événements AWS CloudTrail de données pour obtenir des identifiants de demande

Un compartiment Amazon S3 configuré avec CloudTrail des événements de données pour consigner les opérations d'API au niveau des objets S3 fournit des informations détaillées sur les actions entreprises par un utilisateur, un rôle ou un AWS service dans Amazon S3. Vous pouvez identifier les ID de requête S3 en interrogeant les CloudTrail événements avec Athena.

Utilisation de la journalisation des accès au serveur S3 pour obtenir des ID de demande

Un compartiment Amazon S3 configuré pour journaliser les accès au serveur S3 fournit des enregistrements détaillés des demandes qui lui sont soumises. Vous pouvez identifier les ID de demande S3 en interrogeant les journaux d'accès au serveur à l'aide d'Athena.

Historique du document

- Version de l'API actuelle : 2006-03-01

La table suivante décrit les modifications importantes apportées à chaque version de la Référence d'API Amazon Simple Storage Service et du Guide de l'utilisateur Amazon S3. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
Amazon S3 Inventory prend en charge la clé de Inventory AccessibleOptionalFields condition s3 :	Amazon S3 Inventory prend en charge la clé de Inventory AccessibleOptionalFields condition s3 : pour contrôler si les utilisateurs peuvent inclure des champs de métadonnées facultatifs dans leurs rapports. Pour plus d'informations, consultez la section Création de la configuration du rapport d'inventaire Control S3.	20 février 2024
Support IPv6 pour S3 sur Outposts	Vous pouvez désormais accéder à S3 sur les compartiments Outposts via IPv6 via S3 sur les points de terminaison à double pile d'Outposts. La prise en charge d'IPv6 pour S3 on Outposts vous permet de gérer vos compartiments S3 on Outposts et les ressources du plan de contrôle sur les réseaux IPv6.	16 janvier 2024
Nouvelle classe de stockage Amazon S3 à zone unique	Amazon S3 Express One Zone est une classe de	28 novembre 2023

et hautes performances : S3 Express One Zone	stockage Amazon S3 à zone unique et hautes performances, spécialement conçue pour fournir un accès aux données constant en moins de dix millisecondes pour vos applications les plus sensibles à la latence. Pour plus d'informations, consultez S3 Express One Zone.	
Mountpoint pour Amazon S3 ajoute la prise en charge de S3 Express One Zone	Vous pouvez désormais monter des compartiments de répertoires S3 Express One Zone avec Mountpoint.	28 novembre 2023
Version du schéma d'invocation Lambda	Les opérations par lots Amazon S3 introduisent une nouvelle version du schéma d'invocation Lambda à utiliser avec les tâches d'opérations par lots qui agissent sur des compartiments de répertoires. Pour plus d'informations, consultez Utilisation de Lambda et des opérations par lots Amazon S3 avec des compartiments de répertoires.	28 novembre 2023

Action d’importation pour les compartiments de répertoires	Amazon S3 introduit l’action d’importation. L’importation est une méthode simplifiée pour créer des tâches d’opérations par lots Amazon S3 afin de copier des objets depuis des compartiments à usage général vers des compartiments de répertoires. Pour plus d’informations, consultez Importation d’objets dans un compartiment de répertoires.	28 novembre 2023
Gestion des accès S3 avec les octrois d’accès S3	Amazon S3 Access Grants vous permet de gérer les autorisations de données à grande échelle pour les principaux AWS Identity and Access Management (IAM), en plus des identités d'annuaire issues d'annuaires d'entreprise tels que. Azure AD Vous pouvez désormais appliquer les autorisations S3 de moindre privilège et facilement adapter ces autorisations en fonction des besoins de votre entreprise. Pour plus d’informations, consultez Gestion de l’accès avec les octrois d’accès S3.	26 novembre 2023

Mountpoint pour Amazon S3 ajoute une fonctionnalité de mise en cache	Avec Mountpoint, vous pouvez désormais configurer la mise en cache pour les données consultées à plusieurs reprises.	22 novembre 2023
Génération améliorée de manifestes Amazon S3 Batch Operations	Vous pouvez désormais indiquer aux opérations par lots Amazon S3 de générer automatiquement un manifeste en fonction des critères de filtre d'objet que vous spécifiez lors de la création de votre tâche. Cette option est disponible pour les tâches de réplication par lots que vous créez dans la console Amazon S3, ou pour tout type de tâche que vous créez à l' AWS CLI aide des AWS SDK ou de l'API REST Amazon S3. Pour plus d'informations, consultez Création d'une tâche d'opérations par lots Amazon S3.	22 novembre 2023
Les compartiments Amazon S3 existants peuvent désormais ajouter des configurations de verrouillage d'objet	Vous pouvez désormais activer le verrouillage d'objet sur le compartiment Amazon S3 existant. Vous pouvez définir des mises en suspens juridiques et des périodes de rétention pour les compartiments nouveaux ou existants. Pour plus d'informations, consultez Utilisation du verrouillage des objets.	20 novembre 2023

Métriques de demande S3 Storage Lens pour les préfixes	S3 Storage Lens introduit les métriques de demande pour les préfixes au sein d'un compartiment Amazon S3. Pour plus d'informations, consultez Catégories de métriques.	17 novembre 2023
Groupes Amazon S3 Storage Lens	S3 Storage Lens introduit les groupes Storage Lens, un filtre défini personnalisé pour les objets basé sur les métadonnées des objets. Pour plus d'informations, consultez Utilisation des groupes Amazon S3 Storage Lens.	15 novembre 2023
Nouvelle politique IAM	S3 sur Outposts introduit `AWSServiceRoleForS3OnOutposts` , rôle lié à un service destiné à vous faciliter la gestion des ressources réseau. Pour en savoir plus, consultez Utilisation de rôles liés à un service pour S3 sur Outposts.	3 octobre 2023
Amazon S3 indique l'heure `Last-Modified` pour les marqueurs de suppression	Amazon S3 indique l'heure `Last-Modified` des marqueurs de suppression dans les en-têtes de réponse des opérations `Head` et `Get` de S3 et de l'API, respectiv ement. Pour en savoir plus, consultez Utilisation des marqueurs de suppression.	27 septembre 2023

Amazon S3 : mise à jour de la politique AWS gérée	Amazon S3 a ajouté des autorisations `s3:Describe*` à `AmazonS3ReadOnlyAccess`. Pour plus d'informations, consultez Politiques gérées par AWS pour Amazon S3.	11 août 2023
Temps de démarrage améliorés pour les demandes de restauration standard effectuées via les opérations par lot S3	Les récupérations standard pour les demandes de restauration effectuées via les opérations par lot S3 peuvent désormais démarrer en quelques minutes. Pour plus d'informations, consultez Options de récupération des archives.	9 août 2023
Ajout de Mountpoint, un client à haut débit permettant de monter un compartiment Amazon S3 en tant que système de fichiers local.	Avec Mountpoint, vos applications peuvent accéder aux objets stockés dans Amazon S3 par le biais d'opérations sur les fichiers, ce qui leur permet d'accéder au stockage et au débit élastiques d'Amazon S3 via une interface de fichier.	9 août 2023

Chiffrement double couche côté serveur avec AWS Key Management Service clés (DSSE-KMS)	Le chiffrement double couche côté serveur avec des clés AWS Key Management Service (AWS KMS) (DSSE-KMS) applique deux couches de chiffrement aux objets lorsqu'ils sont chargés sur Amazon S3. Pour plus d'informations, consultez la section Utilisation du chiffrement double couche côté serveur avec des clés. AWS KMS	13 juin 2023
Amazon S3 active la fonctionnalité S3 Bloquer l'accès public et désactive les listes de contrôle d'accès (ACL) S3 pour tous les nouveaux compartiments.	Amazon S3 active désormais automatiquement l'accès public aux blocs S3 et désactive les listes de contrôle d'accès (ACL) S3 pour tous les nouveaux compartiments S3 dans toutes les régions. AWS Pour en savoir plus, consultez Blocage de l'accès public à votre stockage Amazon S3 et Contrôle de la propriétés des objets et désactivation des listes ACL pour votre compartiment.	27 avril 2023
Métrique d'échec des opérations de réplication S3	Amazon S3 ajoute une nouvelle Amazon CloudWatch métrique pour surveiller les échecs de réplication S3. Pour plus d'informations, consultez Surveillance de l'avancement des métriques de réplication.	5 avril 2023

DNS privé	AWS PrivateLink pour Amazon S3 prend désormais en charge le DNS privé. Pour plus d'informations, consultez Private DNS (DNS privés).	14 mars 2023
Prise en charge des points d'accès intercompte dans la console Amazon S3	Amazon S3 prend désormais en charge la création de points d'accès intercompte à l'aide de la console Amazon S3. Pour plus d'informations, consultez Création de points d'accès.	14 mars 2023
Amazon S3 sur Outposts prend en charge la réplication S3 sur Outposts	Avec la réplication S3 locale, vous pouvez automatiquement répliquer des objets vers un compartiment de destination Outposts unique ou plusieurs compartiments de destination. Les compartiments de destination peuvent se trouver dans des Outposts différents AWS Outposts ou dans les mêmes Outposts que le compartiment source. Pour plus d'informations, consultez Replicating objects for S3 sur Outposts (Réplication d'objets pour S3 sur Outposts).	14 mars 2023

Alias des points d'accès Amazon S3 Object Lambda	Quand vous créez un point d'accès Object Lambda, Amazon S3 génère automatiquement un alias unique pour votre point d'accès Object Lambda. Vous pouvez utiliser cet alias à la place d'un nom de compartiment Amazon S3 ou de l'Amazon Resource Name (ARN) du point d'accès Object Lambda dans une demande pour les opérations de plan de données de point d'accès. Pour plus d'informations, consultez How to use a bucket-style alias for your Object Lambda Access Point (Comment utiliser un alias de type compartiment pour votre point d'accès Object Lambda).	14 mars 2023
Prise en charge des points d'accès multi-régions intercomptes d'Amazon S3	Amazon S3 prend désormais en charge la création de points d'accès multirégions intercomptes à l'aide de la console Amazon S3. Pour plus d'informations, consultez Création de points d'accès multi-Régions.	14 mars 2023

Points d'accès intercompte	Amazon S3 prend en charge la création de points d'accès intercompte. Vous pouvez créer un point d'accès intercompte en utilisant l'opération AWS Command Line Interface (AWS CLI) ou l'opération d'API REST `CreateAccessPoint` . Pour plus d'informations, consultez Création de points d'accès.	30 novembre 2022
Amazon S3 prend en charge les contrôles de basculement pour les points d'accès multi-régions d'Amazon S3	Amazon S3 introduit le contrôle du basculement pour les points d'accès multi-régions. Ces contrôles vous permettent de déplacer le trafic des requêtes d'accès aux données S3 acheminées par un point d'accès multi-régions Amazon S3 vers une autre Région AWS en quelques minutes pour tester et créer des applications hautement disponibles. Pour plus d'informations, reportez-vous à Amazon S3 Multi-Region Access Point failover controls (Contrôles du basculement du point d'accès multi-régions d'Amazon S3).	28 novembre 2022

Amazon S3 Storage Lens améliore la visibilité de l'organisation grâce à 34 nouvelles métriques	S3 Storage Lens introduit 34 métriques supplémentaires pour découvrir des opportunités d'optimisation des coûts plus approfondies, identifier les bonnes pratiques de protection des données et améliorer les performances des flux d'applications. Pour plus d'informations, consultez S3 Storage Lens metrics (Métriques S3 Storage Lens).	17 novembre 2022
Amazon S3 prend en charge des taux de demande de restauration plus élevés pour S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive	Amazon S3 prend en charge les demandes de restauration à un taux pouvant atteindre 1 000 transactions par seconde, Compte AWS pour les classes de stockage S3 Glacier Flexible Retrieval et S3 Glacier Deep Archive.	15 novembre 2022

Amazon S3 sur Outposts prend en charge des actions et des filtres de cycle de vie S3 supplémentaires	S3 sur Outposts prend en charge des règles de cycle de vie S3 supplémentaires pour optimiser la gestion de la capacité. Vous pouvez faire expirer des objets lorsqu'ils vieillissent ou sont remplacés par des versions plus récentes. Vous pouvez créer une règle de cycle de vie pour un compartiment entier ou un sous-ensemble d'objets dans un compartiment en filtrant avec des préfixes, des étiquettes d'objets ou la taille des objets. Pour plus d'informations, consultez Création et gestion d'une configuration de cycle de vie.	2 novembre 2022
Prise en charge de la réplication S3 pour les objets SSE-C	Vous pouvez répliquer des objets qui sont créés à l'aide d'un chiffrement côté serveur avec des clés fournies par le client. Pour plus d'informations sur la réplication d'objets chiffrés, consultez Replicating objects created with server-side encryption (SSE-C, SSE-S3, SSE-KMS) [Réplication d'objets créés avec le chiffrement côté serveur (SSE-C, SSE-S3, SSE-KMS)].	24 octobre 2022

Amazon S3 sur Outposts prend en charge les alias de point d'accès	Avec S3 on Outposts, vous devez utiliser des points d'accès pour accéder à tout objet dans un compartiment Outposts. Chaque fois que vous créez un point d'accès pour un compartiment, S3 sur Outposts génère automatiquement un alias de point d'accès. Vous pouvez utiliser cet alias de point d'accès plutôt qu'un ARN de point d'accès pour toutes les opérations de plan de données. Pour plus d'informations, consultez Utilisation d'un alias de type compartiment pour votre point d'accès S3 on Outposts.	21 octobre 2022
S3 Object Lambda prend en charge les opérations `HeadObject` , `ListObjects` et `ListObjectsV2`	Vous pouvez utiliser du code personnalisé pour modifier les données renvoyées par des requêtes `GET`, `LIST` ou `HEAD` S3 standard afin de filtrer les lignes, de redimensionner les images de manière dynamique, de supprimer des données confidentielles et plus encore. Pour en savoir plus, consultez Transformation d'objets avec S3 Object Lambda.	4 octobre 2022

Amazon S3 sur Outposts prend en charge la gestion des versions S3	Une fois activé, la gestion des versions S3 enregistre plusieurs copies différentes d'un objet dans le même compartiment. Vous pouvez utiliser la gestion des versions S3 pour préserver, récupérer et restaurer chaque version de chaque objet stocké dans votre compartiment Outposts. La gestion des versions S3 vous aide à récupérer en cas d'action involontaire d'un utilisateur et de défaillance applicative. Pour plus d'informations, consultez Gestion de la gestion des versions S3 pour votre compartiment S3 on Outposts.	21 septembre 2022
AWS Backup pour Amazon S3	AWS Backup est un service entièrement géré basé sur des politiques que vous pouvez utiliser pour définir une politique de sauvegarde centralisée afin de protéger vos données Amazon S3. Pour plus d'informations, consultez Utilisation AWS Backup pour Amazon S3.	18 février 2022

Utiliser la réplication par lot S3 pour répliquer des objets existants	Avec la réplication par lot S3, vous pouvez répliquer des objets qui existaient avant la mise en place d'une configuration de réplication. La réplication d'objets existants se fait par l'utilisation d'une tâche d'opérations par lot. La réplication par lot S3 diffère de la réplication en direct qui copie en continu et automatiquement les nouveaux objets d'un compartiment Amazon S3 à l'autre. Pour plus d'informations, consultez Réplication d'objets existants avec la réplication par lot S3.	8 février 2022
Changement de nom de S3 Glacier Flexible Retrieval	La classe de stockage Glacier a été renommée S3 Glacier Flexible Retrieval. Cette modification n'a aucune répercussion sur l'API.	30 novembre 2021

Nouveau paramètre S3 Object Ownership pour désactiver les listes ACL	Vous pouvez appliquer le paramètre appliqué par le propriétaire du compartiment pour Object Ownership afin de désactiver les listes ACL pour votre compartiment et les objets qu'il contient et de prendre possession de chaque objet de votre compartiment. Le paramètre appliqué par le propriétaire du compartim ent simplifie la gestion des accès aux données stockées dans Amazon S3. Pour en savoir plus, consultez Contrôle de la propriété des objets et désactivation des listes ACL pour votre compartiment.	30 novembre 2021
Nouvelle classe de stockage S3 Intelligent-Tiering	L'accès S3 Intelligent-Tiering Archive Instant est une classe de stockage supplémentaire sous S3 Intelligent-Tierin g. Pour plus d'informations, consultez Fonctionnement de S3 Intelligent-Tiering.	30 novembre 2021
Nouvelle classe de stockage S3 Glacier Instant Retrieval	Vous pouvez désormais placer des objets dans la classe de stockage S3 Glacier Instant Retrieval. Pour en savoir plus sur cette classe de stockage, utilisez Utilisation des classes de stockage Amazon S3.	30 novembre 2021

AWS Backup pour Amazon S3 Preview	AWS Backup est un service entièrement géré basé sur des politiques que vous pouvez utiliser pour définir une politique de sauvegarde centralisée afin de protéger vos données Amazon S3. Pour plus d'informations, consultez Utilisation AWS Backup pour Amazon S3.	30 novembre 2021
AWS Identity and Access Management Access Analyzer pour Amazon S3	IAM Access Analyzer exécute des vérifications de stratégie s pour valider votre stratégie par rapport à la grammaire de stratégie et aux bonnes pratiques IAM. Pour en savoir plus sur la validatio n des stratégies à l'aide d'IAM Access Analyzer, consultez Validation de stratégie IAM Access Analyzer dans le Guide de l'utilisa teur IAM.	30 novembre 2021
Nouveaux types d'événements	Nouveaux types d'événeme nts ajoutés aux notifications d'événements Amazon S3. Consultez Notifications d'événements Amazon S3.	29 novembre 2021
Activez Amazon EventBridge sur des buckets	Vous pouvez activer EventBrid ge les compartiments Amazon S3 pour envoyer des événements à Amazon EventBridge, voir Utilisation EventBridge.	29 novembre 2021

Nouveaux filtres de cycle de vie S3	Vous pouvez créer des règles de cycle de vie basées sur la taille de l'objet ou spécifier le nombre de versions non courantes de l'objet à conserver. Pour plus d'informations, consultez Examples of S3 Lifecycle configuration (Exemples de configuration du cycle de vie S3).	23 novembre 2021
Publier les métriques d'Amazon S3 Storage Lens sur Amazon CloudWatch	Vous pouvez publier les indicateurs d'utilisation et d'activité de S3 Storage Lens sur Amazon CloudWatch afin de créer une vue unifiée de votre santé opérationnelle dans des CloudWatch tableaux de bord. Vous pouvez également utiliser des CloudWatch fonctionnalités, telles que les alarmes et les actions déclenchées, les mathématiques métriques et la détection des anomalies, pour surveiller les métriques S3 Storage Lens et prendre des mesures en conséquence. En outre, les CloudWatch API permettent aux applications, y compris aux fournisseurs tiers, d'accéder à vos métriques S3 Storage Lens. Pour plus d'informations, consultez les métriques Monitor S3 Storage Lens dans CloudWatch.	22 novembre 2021

Points d'accès multi-Régions	Vous pouvez utiliser des points d'accès multi-régions pour créer un point de terminaison global que les applications peuvent utiliser pour traiter les demandes provenant de compartiments Amazon S3 situés dans plusieurs Régions AWS. Vous pouvez utiliser ce point d'accès multi-Régions pour acheminer les données vers un compartiment présentant la latence la plus faible. Pour en savoir plus sur les points d'accès multi-régions et leur utilisation, consultez Point d'accès multi-régions dans Simple Storage Service (Amazon S3).	2 septembre 2021

Simple Storage Service (Amazon S3) on Outposts ajoute un accès local direct aux applications	Exécutez vos applications en dehors du cloud privé AWS Outposts virtuel (VPC) et accédez à vos données S3 on Outposts. Vous pouvez également accéder aux objets S3 sur Outposts directeme nt depuis votre réseau sur site. Pour en savoir plus sur la configuration des points de terminaison S3 sur Outposts à l'aide des adresses IP clients (CoIP) et l'accès à vos objets en créant une passerelle locale à partir de votre réseau sur site, consultez la section Accès à Amazon S3 sur Outposts à l'aide de points d'accès VPC uniquement.	29 juillet 2021

Alias de points d'accès Amazon S3	Lorsque vous créez un point d'accès, Amazon S3 génère automatiquement un alias que vous pouvez utiliser pour l'accès aux données au lieu d'un nom de compartiment. Vous pouvez utiliser cet alias de point d'accès plutôt qu'un Amazon Resource Name (ARN) pour toutes les opérations de plan de données de point d'accès. Pour en savoir plus, veuillez consulter la section Using a bucket-style alias for your access point (Utilisation d'un alias de type compartiment pour votre point d'accès).	26 juillet 2021
Amazon S3 Inventory et les opérations par lot S3 prennent en charge le statut de la clé de compartiment S3	L'inventaire et les opérations par lot Amazon S3 prennent en charge l'identification et la copie d'objets existants avec des clés de compartiment S3 Les clés de compartiment S3 accélèrent la réduction des coûts de chiffrement côté serveur pour des objets existants. Pour plus d'informations, consultez Inventaire Amazon S3 et Copie d'objet à l'aide d'opérations par lot.	3 juin 2021

Instantané du compte des métriques d'Simple Storage Service (Amazon S3) Storage Lens	L'instantané du compte S3 Storage Lens affiche votre stockage total, le nombre d'objets et la taille moyenne des objets sur la page d'accueil de la console S3 (Buckets (Compartiments) en résumant les métriques de votre tableau de bord par défaut. Pour plus d'informations, consultez Instantané du compte des métriques d'Amazon S3 Storage Lens.	5 mai 2021
Prise en charge accrue des points de terminaison Simple Storage Service (Amazon S3) on Outposts	S3 on Outposts prend désormais en charge jusqu'à 100 points de terminaison par Outpost. Pour plus d'informations, consultez Restrictions réseau de S3 on Outposts.	29 avril 2021
Notifications d'événements Amazon S3 on Outposts dans Amazon Events CloudWatch	Vous pouvez utiliser CloudWatch Events pour créer une règle afin de capturer n'importe quel événement de l'API S3 on Outposts et d'être averti par le biais de toutes les cibles prises en charge CloudWatch . Pour plus d'informations, consultez la section Recevoir des notifications d'événements S3 on Outposts à l'aide d' CloudWatch événements.	19 avril 2021

S3 Object Lambda	S3 Object Lambda vous donne la possibilité d'intégrer votre propre code aux requêtes GET Amazon S3 afin de modifier et de traiter les données lorsqu'elles sont renvoyées vers une application. Vous pouvez utiliser du code personnalisé pour modifier les données renvoyées par des requêtes GET S3 standard afin de filtrer les lignes, de redimensionner les images de manière dynamique, de supprimer des données confidentielles et plus encore. Pour plus d'informations, voir Transformation d'objets.	18 mars 2021
AWS PrivateLink	Avec AWS PrivateLink Amazon S3, vous pouvez vous connecter directement à S3 en utilisant un point de terminaison d'interface dans votre cloud privé virtuel (VPC) au lieu de vous connecter via Internet. Les points de terminaison d'interface sont directement accessibles à partir des applications sur site ou dans une Région AWS différente. Pour plus d'informations, consultez Technologie AWS PrivateLink pour Amazon S3.	2 février 2021

Gérer la capacité d'Amazon S3 on Outposts avec AWS CloudTrail	Les événements de gestion de S3 on Outposts sont disponibles via CloudTrail les journaux. Pour plus d'informations, consultez Gérer la capacité de S3 on Outposts avec. CloudTrail	21 décembre 2020
Forte cohérence	Amazon S3 assure une forte read-after-write cohérence pour l'ensemble des objets de votre compartiment S3 `PUT` et des `DELETE` requêtes y afférentes Régions AWS. En outre, les opérations de lecture sur Amazon S3 Select, les listes de contrôle d'accès Amazon S3, les balises d'objet Amazon S3 et les métadonnées d'objet (par exemple, l'objet `HEAD`) sont fortement cohérentes. Pour plus d'informations, consultez Modèle de cohérence de données Amazon S3.	1er décembre 2020

Synchronisation des modifications de réplica Amazon S3	La synchronisation des modifications de réplica Amazon S3 permet de synchroniser les métadonnées d'objet telles que les balises, les listes ACL et les paramètres de verrouillage des objets entre les objets sources et les réplicas. Lorsque cette fonctionnalité est activée, Amazon S3 réplique les modifications de métadonnées apportées à l'objet source ou aux copies de réplica. Pour plus d'informations, consultez la Réplication des modifications de métadonnées avec synchronisation des modifications de réplica.	1er décembre 2020

Clés de compartiment Amazon S3	Les clés de compartiment Amazon S3 réduisent le coût du chiffrement côté serveur Amazon S3 à l'aide d' AWS Key Management Service (SSE-KMS). Cette nouvelle clé au niveau du compartiment pour le chiffrement côté serveur peut réduire les coûts des demandes AWS KMS jusqu'à 99 % en diminuant le trafic de demandes d'Amazon S3 vers AWS KMS. Pour plus d'informations, consultez la section Réduire le coût du SSE-KMS à l'aide des clés de compartiment S3.	1er décembre 2020

Cadre de stockage Amazon S3	S3 Storage Lens regroupe vos métriques et affiche les informations dans la section Account snapshot (Instantané du compte) sur la page Buckets (Compartiments) de la console Amazon S3. S3 Storage Lens fournit également un tableau de bord interactif que vous pouvez utiliser pour visualiser les informations et les tendances, signaler les anomalies et recevoir des recommandations pour optimiser les coûts de stockage et appliquer les bonnes pratiques de protection des données. Votre tableau de bord dispose d'options d'exploration pour générer et visualiser des informations au niveau de l'organisation, du compte, de la Région AWS, de la classe de stockage, du compartiment, du préfixe ou du groupe Storage Lens. Vous pouvez également envoyer une exportation de métriques quotidienne au format CSV ou Parquet vers un compartiment S3. Pour plus d'informations, consultez Évaluer l'activité et l'utilisation de votre stockage avec S3 Storage Lens.	18 novembre 2020

Suivi des requêtes S3 à l'aide de AWS X-Ray	Amazon S3 s'intègre à X-Ray pour propager le contexte de suivi et vous donner une chaîne de demandes avec des nœuds en amont et en aval. Pour plus d'informations, consultez Suivi des demandes à l'aide de X-Ray.	16 novembre 2020
Métriques de réplication S3	Les métriques de réplication S3 fournissent des indicateurs détaillés pour les règles de réplication dans votre configuration de réplication. Pour plus d'informations, consultez Métriques de réplication et notifications d'événements Amazon S3.	9 novembre 2020
Accès aux archives S3 Intelligent-Tiering (Hiérarchisation intelligente) et accès à Deep Archive	L'accès aux archives S3 Intelligent-Tiering (Hiérarchisation intelligente) et l'accès à Deep Archive sont des niveaux de stockage supplémentaires dans le cadre de S3 Intelligent-Tiering (Hiérarchisation intelligente). Pour plus d'informations, consultez la Classe de stockage pour optimiser automatiquement les objets fréquemment et rarement consultés.	9 novembre 2020

Réplication du marqueur de suppression	Avec la réplication du marqueur de suppression, vous pouvez garantir que les marqueurs de suppressi on sont copiés dans vos compartiments cible pour vos règles de réplication. Pour plus d'informations, consultez Utilisation de la réplication des marqueurs de suppression.	9 novembre 2020
Propriété de l'objet S3	La propriété de l'objet est un paramètre de compartiment S3 que vous pouvez utiliser pour contrôler la propriété des nouveaux objets qui sont téléchargés dans vos compartiments. Pour plus d'informations, consultez Utilisation de la propriété de l'objet S3.	2 octobre 2020

Amazon S3 on Outposts	Avec Amazon S3 on Outposts, vous pouvez créer des compartiments S3 sur vos AWS Outposts ressources et stocker et récupérer facilement des objets sur site pour les applications qui nécessitent un accès aux données locales, un traitement local des données et une résidence des données. Vous pouvez utiliser S3 sur Outposts via les AWS Management Console AWS SDK ou AWS CLI l'API REST. Pour plus d'informations, consultez Utilisation d'Amazon S3 sur Outposts.	30 septembre 2020
Condition propriétaire du compartiment	Vous pouvez utiliser la condition de propriétaire du compartiment Amazon S3 pour vous assurer que les compartiments Comptes AWS que vous utilisez dans vos opérations S3 correspondent à vos attentes. Pour plus d'informations, consultez Condition propriétaire du compartiment.	11 septembre 2020

Prise en charge des opérations par lot S3 pour la rétention du verrouillage d'objet	Vous pouvez désormais utiliser les opérations par lot avec le verrouillage d'objet S3 pour appliquer des paramètres de conservation à de nombreux objets Simple Storage Service (Amazon S3) à la fois. Pour plus d'informations, consultez Définition des dates de rétention du verrouillage d'objets S3 à l'aide des opérations par lot S3.	4 mai 2020
Prise en charge des opérations par lot S3 pour le blocage juridique du verrouillage d'objet	Vous pouvez désormais utiliser les opérations par lot avec le verrouillage d'objets S3 pour ajouter un blocage juridique à de nombreux objets Amazon S3 à la fois. Pour plus d'informations, consultez Utilisation des opérations par lot S3 pour définir la mise en suspens juridique du verrouillage des objets S3.	4 mai 2020
Balises de tâche pour les opérations par lot S3	Vous pouvez ajouter des balises à vos tâches d'opérations par lot S3 pour contrôler et étiqueter ces tâches. Pour plus d'informations, consultez Balises pour les tâches d'opérations par lot S3.	16 mars 2020

Points d'accès Amazon S3	Les points d'accès Amazon S3 simplifient la gestion de l'accès aux données à grande échelle pour les ensembles de données partagés dans S3. Les points d'accès sont des points de terminaison réseau associés à des compartiments que vous pouvez utiliser pour effectuer des opérations d'objet S3. Pour plus d'informations, veuillez consulter la rubrique Gestion de l'accès aux données à l'aide des points d'accès Amazon S3.	2 décembre 2019
Analyseur d'accès pour Simple Storage Service (Amazon S3)	Access Analyzer pour Amazon S3 vous avertit de la présence de compartiments S3 configurés pour autoriser l'accès à toute personne sur Internet ou autre Comptes AWS, y compris à des comptes extérieurs à votre organisation. Pour plus d'informations, consultez Utilisation de l'analyseur d'accès pour Amazon S3.	2 décembre 2019

Contrôle du délai de réplication S3 (S3 RTC)	Le contrôle du délai de réplication S3 (S3 RTC) réplique la plupart des objets téléchargés sur Amazon S3 en quelques secondes, et 99,99 % de ces objets dans les 15 minutes. Pour plus d'informations, consultez Réplication d'objets à l'aide du contrôle du temps de réplication S3 (S3 RTC).	20 novembre 2019
Réplication dans une même Région	Vous pouvez utiliser la réplication dans une même région (SRR) pour copier des objets entre compartiments Amazon S3 d'une même Région AWS. Pour plus d'informations sur la réplication entre Régions et la réplication dans une même région, consultez Réplication.	18 septembre 2019
Prise en charge de la réplication entre Régions pour le verrouillage d'objet S3	La réplication entre Régions prend désormais en charge le verrouillage d'objet. Pour plus d'informations, consultez Qu'est-ce qu'Amazon S3 réplique ?.	28 mai 2019

Opérations par lot S3	À l'aide des opérations par lot S3, vous pouvez effectuer des opérations par lot à grande échelle sur des objets Amazon S3. Les opérations par lot S3 peuvent exécuter une seule opération sur les listes d'objets que vous spécifiez. Un même travail peut effectuer l'opération spécifiée sur des milliards d'objets contenant des exaoctets de données. Pour plus d'informations, consultez la section Exécution des Opérations par lot S3.	30 avril 2019
Région Asie-Pacifique (Hong Kong)	Amazon S3 est désormais disponible dans la Région Asie-Pacifique (Hong Kong). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	24, 2019 avril 2019
Nouveau champ ajouté aux journaux d'accès au serveur	Amazon S3 a ajouté le nouveau champ suivant aux journaux d'accès au serveur : version de protocole TLS (Transport Layer Security) Pour plus d'informations, consultez Format des journaux d'accès au serveur.	28 mars 2019

Nouvelle classe de stockage d'archive	Amazon S3 propose désormais une nouvelle classe de stockage d'archive , S3 Glacier Deep Archive (`DEEP_ARCHIVE`), pour le stockage d'objets rarement consultés. Pour plus d'informations, consultez Classes de stockage.	27 mars 2019
Nouveaux champs ajoutés aux journaux d'accès au serveur	Amazon S3 a ajouté les nouveaux champs suivants aux journaux d'accès au serveur : Host Id, Signature Version, Cipher Suite, Authentication Type et Host Header. Pour plus d'informations, consultez Format des journaux d'accès au serveur.	5 mars 2019
Prise en charge des fichiers Amazon S3 Inventory au format Parquet	Simple Storage Service (Amazon S3) prend désormais en charge le format Apache Parquet (Parquet) en plus du format Apache ORC (Optimized Row Columnar) et du format de fichier CSV (valeurs séparées par une virgule) pour les fichiers de sortie d'inventaire. Pour plus d'informations, consultez Inventaire.	4 décembre 2018

Verrouillage des objets S3	Amazon S3 prend désormais en charge la fonctionnalité de verrouillage d'objets qui fournit les protections Write Once Read Many (WORM) pour les objets Amazon S3. Pour plus d'informations, consultez Verrouillage des objets.	26 novembre 2018
Mise à niveau de la vitesse de restauration	À l'aide de la mise à niveau de la vitesse de restauration Amazon S3, vous pouvez accélérer la vitesse d'une restauration depuis la classe de stockage S3 Glacier Flexible Retrieval pendant que la restauration est en cours. Pour plus d'informations, consultez Restauration des objets archivés.	26 novembre 2018
Notifications d'événement de restauration	Les notifications d'événement Amazon S3 prennent désormais en charge le lancement et l'arrêt des événements lors de la restauration des objets depuis la classe de stockage S3 Glacier Flexible Retrieval. Pour plus d'informations, consultez Notifications d'événement.	26 novembre 2018

PUT directement sur la classe de stockage S3 Glacier Flexible Retrieval	L'opération PUT Amazon S3 prend désormais en charge la spécification S3 Glacier Flexible Retrieval comme classe de stockage lorsque vous créez des objets. Avant, vous deviez transférer les objets vers la classe de stockage S3 Glacier Flexible Retrieval à partir d'une autre classe de stockage Amazon S3. De même, lorsque vous utilisez la réplication entre Régions S3, vous pouvez désormais spécifier S3 Glacier Flexible Retrieval comme classe de stockage pour les objets répliqués . Pour en savoir plus sur la classe de stockage S3 Glacier Flexible Retrieval , consultez Classes de stockage. Pour en savoir plus sur la spécification de la classe de stockage pour les objets répliqués, consultez Présentation de la configuration de réplication. Pour plus d'informations sur la modification de la commande directe PUT sur l'API REST S3 Glacier Flexible Retrieval , consultez Historique du document :	26 novembre 2018

	PUT directement sur S3 Glacier Flexible Retrieval.	
Nouvelle classe de stockage	Amazon S3 propose une nouvelle classe de stockage nommée S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) conçue pour les données de longue durée avec des modèles d'accès variables ou inconnus. Pour plus d'informations, consultez Classes de stockage.	26 novembre 2018
Blocage de l'accès public Amazon S3	Amazon S3 inclut désormais la possibilité de bloquer l'accès public aux compartiments et aux objets par compartiment ou à l'échelle d'un compte. Pour plus d'informations, consultez Utilisation de la fonctionnalité de blocage de l'accès public Amazon S3.	15 novembre 2018

Filtrage des améliorations dans les règles de la réplication entre Régions (CRR	Dans une configuration de règle CRR, vous pouvez spécifier un filtre d'objet pour choisir un sous-ensemble d'objets auquel s'applique la règle. Précédemment, vous ne pouviez filtrer que sur un préfixe de clé d'objet. Dans cette version, vous pouvez filtrer sur un préfixe de clé d'objet, une ou plusieurs balises d'objet, ou les deux. Pour plus d'informations, consultez la Configuration CRR : présentation de la configuration de réplication.	19 septembre 2018
Nouvelles fonctions d'Simple Storage Service (Amazon S3) Select	Amazon S3 Select prend désormais en charge les entrées Apache Parquet, les requêtes sur des objets JSON imbriqués et deux nouvelles métriques CloudWatch de surveillance Amazon (`SelectScannedBytes` et `SelectReturnedBytes`).	5 septembre 2018
Mises à jour disponibles sur RSS	Vous pouvez à présent vous abonner à un flux RSS pour recevoir les notifications des mises à jour du Guide d'utilisateur Amazon S3.	19 juin 2018

Mises à jour antérieures

Le tableau ci-après décrit les modifications importantes apportées dans chaque version du Guide de l'utilisateur Amazon S3 avant le 19 juin 2018.

Modification	Description	Date
Mise à jour d'exemples de code	Exemples de code mis à jour : • C# - Mise à jour de l'ensemble des exemples afin d'utiliser le modèle asynchrone basé sur tâche. Pour plus d'informations, consultez les API asynchrones d'Amazon Web Services pour .NET dans le Guide du AWS SDK for .NET développeur. Les exemples de code fournis sont désormais compatibles avec la version 3 du kit AWS SDK for .NET. • Java - Mise à jour de l'ensemble des exemples afin d'utiliser le modèle de générateur client. Pour plus d'informations sur le modèle de générateur client, consultez Création de clients de service. • PHP - Mise à jour de l'ensemble des exemples afin d'utiliser le kit AWS SDK for PHP 3.0. Pour plus d'informations sur la AWS SDK for PHP version 3.0, consultez AWS SDK for PHP. • Ruby : exemple de code mis à jour afin que les exemples fonctionnent avec la AWS SDK for Ruby version 3.	30 avril 2018
Amazon S3 indique désormais les classes de récupération et de `ONEZONE_IA` stockage flexibles de S3 Glacier dans les métriques de stockage	Outre les octets effectifs, ces métriques de stockage comprennent les octets de supplément par objet pour les classes de stockage applicables (`ONEZONE_IA` , `STANDARD_IA` et S3 Glacier Flexible Retrieval) : •	30 avril 2018

Modification	Description	Date
d'Amazon CloudWatch Logs	Pour les objets de classe de stockage `ONEZONE_IA` et `STANDARD_IA`, Simple Storage Service (Amazon S3) rapporte les objets de taille inférieure à 128 Ko comme ayant une taille de 128 Ko. Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). • Pour les objets de classe de stockage S3 Glacier Flexible Retrieval, les métriques de stockage rapportent les suppléments suivants : • Un supplément de 32 Ko par objet, facturé au tarif de la classe de stockage S3 Glacier Flexible Retrieval • Un supplément de 8 Ko par objet, facturé au tarif de la classe de stockage `STANDARD` Pour plus d'informations, consultez Transition des objets à l'aide du cycle de vie Amazon S3. Pour plus d'informations sur les métriques de stockage, consultez Surveillance des métriques avec Amazon CloudWatch.	
Nouvelle classe de stockage	Amazon S3 propose désormais une nouvelle classe de stockage, `STANDARD_IA` (où « IA » signifie « infrequent access », ou accès peu fréquent) pour le stockage d'objets. Cette classe de stockage est optimisée pour les données à longue durée de vie et moins fréquemment consultées. Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3).	4 avril 2018

Modification	Description	Date
Amazon S3 Select	Amazon S3 prend désormais en charge la récupération de contenu d'objet basée sur une expression SQL. Pour plus d'informations, consultez Filtrer et récupérer des données à l'aide d'Amazon S3 Select.	4 avril 2018
Région Asie-Pacifique (Osaka-Local)	Amazon S3 est maintenant disponible dans la Région Asie-Pacifique (Osaka-Local). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS. **Important** Vous pouvez utiliser la Région Asie-Pacifique (Osaka-Local) uniquement en conjonction avec la Région Asie-Pacifique (Tokyo). Pour demander l'accès à la Région Asie-Pacifique (Osaka-Local), veuillez contacter votre représentant commercial.	12 février 2018
Horodatage de création d'Amazon S3 Inventory	Amazon S3 Inventory comprend désormais un horodatage de la date et de l'heure du début de la création du rapport Amazon S3 Inventory. Vous pouvez utiliser l'horodatage pour déterminer les modifications dans votre stockage Amazon S3 à partir de la date de début ou lorsque le rapport d'inventaire a été généré.	16 janvier 2018
Région Europe (Paris)	Amazon S3 est désormais disponible dans la Région Europe (Paris). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	18 décembre 2017
Région Chine (Ningxia)	Amazon S3 est désormais disponible dans la Région Chine (Ningxia) Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	29 novembre 2017

Modification	Description	Date
Prise en charge des fichiers Amazon S3 Inventory au format ORC	Amazon S3 prend désormais en charge le format Apache ORC (Optimized Row Columnar) en plus du format de fichier CSV (valeurs séparées par une virgule) pour les fichiers de sortie d'inventaire. Vous pouvez également interroger l'inventaire Amazon S3 en utilisant SQL standard à l'aide d'Amazon Athena, Amazon Redshift Spectrum et d'autres outils tels que Presto, Apache Hive et Apache Spark. Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).	17 novembre 2017
Chiffrement par défaut pour les compartiments S3	Le chiffrement par défaut d'Amazon S3 permet de définir le comportement de chiffrement par défaut pour un compartiment S3. Vous pouvez définir le chiffrement par défaut sur un compartiment afin que tous les objets soient chiffrés lorsqu'ils sont stockés dans le compartiment. Les objets sont chiffrés à l'aide du chiffrement côté serveur avec des clés gérées Amazon S3 (SSE-S3) ou des clés AWS gérées (SSE-KMS). Pour plus d'informations, consultez Définition du comportement de chiffrement côté serveur par défaut pour les compartiments Amazon S3.	06 novembre 2017
Statut de chiffrement dans Amazon S3 Inventory	Amazon S3 prend désormais en charge l'inclusion du statut de chiffrement dans Amazon S3 Inventory afin que vous puissiez voir comment vos objets sont chiffrés au repos pour des audits de conformité ou à d'autres fins. Vous pouvez également configurer le chiffrement d'Amazon S3 Inventory avec le chiffrement côté serveur (SSE) ou SSE-KMS afin que tous les fichiers d'inventaire soient chiffrés en conséquence. Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).	06 novembre 2017

Modification	Description	Date
Améliorations de la réplication entre Régions (CRR)	La réplication entre Régions prend désormais en charge les aspects suivants : • Dans un scénario à plusieurs comptes, vous pouvez ajouter une configuration CRR pour remplacer le propriétaire des réplicas par le Compte AWS qui détient le compartiment de destination. Pour plus d'informations, consultez Modification du propriétaire d'un réplica. • Par défaut, Amazon S3 ne réplique pas les objets de votre compartiment source créés à l'aide du chiffrement côté serveur à l'aide de clés stockées AWS KMS dans votre configuration CRR, vous pouvez désormais demander à Amazon S3 de répliquer ces objets. Pour plus d'informations, consultez Réplication d'objets chiffrés (SSE-C, SSE-S3, SSE-KMS, DSSE-KMS).	06 novembre 2017
Europe (London) Region	Amazon S3 est désormais disponible dans la Région Europe (Londres). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	13 décembre 2016
Canada (Central) Region	Amazon S3 désormais disponible dans la Région Canada (Centre) Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	8 décembre 2016

Modification	Description	Date
Balisage des objets	Amazon S3 prend désormais en charge le balisage des objets. Le balisage des objets vous permet de classer le stockage par catégorie. Les préfixes des noms de clés d'objet vous permettent également de classer le stockage par catégorie ; le balisage des objets y ajoute une autre dimension. Il existe des avantages supplémentaires offerts par le balisage. Il s'agit des licences suivantes : • Les balises d'objets permettent un contrôle d'accès précis des autorisations (par exemple, vous pouvez octroyer des autorisations aux utilisateurs IAM pour lire uniquement des objets avec des balises spécifiques). • Contrôle précis en spécifiant la configuration du cycle de vie. Vous pouvez spécifier des balises pour sélectionner un sous-ensemble d'objets auxquels la règle de cycle de vie s'applique. • Si vous avez configuré la réplication entre Régions (CRR), Amazon S3 peut répliquer les balises. Vous devez octroyer les autorisations nécessaires au rôle IAM assumé par Amazon S3 pour répliquer des objets en votre nom. • Vous pouvez également personnaliser CloudWatch les mesures et les CloudTrail événements pour afficher les informations par le biais de filtres de balises spécifiques. Pour plus d’informations, consultez Catégorisation de votre stockage à l'aide de balises.	29 novembre 2016

Modification	Description	Date
Le cycle de vie Amazon S3 prend désormais en charge les filtres basés sur les balises.	Amazon S3 prend désormais en charge le filtrage basé sur des balises dans la configuration du cycle de vie. Vous avez la possibilité de spécifier les règles de cycle de vie dans lesquelles vous pouvez indiquer un préfixe de clé, une ou plusieurs balises d'objets, ou une combinaison des deux pour sélectionner un sous-ensemble d'objets auquel la règle de cycle de vie s'applique. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.	29 novembre 2016
CloudWatch mesures de demande pour les buckets	Amazon S3 prend désormais en charge CloudWatch les métriques pour les demandes effectuées sur des buckets. Lorsque vous activez ces métriques pour un compartiment, les métriques fournissent des résultats toutes les minutes. Vous pouvez également configurer les objets d'un compartiment qui signaleront ces métriques de demandes. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.	29 novembre 2016
Inventaire Amazon S3	Amazon S3 prend désormais en charge l'inventaire de stockage. Amazon S3 Inventory fournit une sortie de fichiers plats de vos objets et leurs métadonnées correspondantes sur une base quotidienne ou hebdomadaire pour un compartiment S3 ou un préfixe partagé (c'est-à-dire, les objets qui ont des noms qui commencent par une chaîne commune). Pour plus d'informations, consultez Inventaire Simple Storage Service (Amazon S3).	29 novembre 2016

Modification	Description	Date
Analyses Amazon S3 – Analyse de classe de stockage	La nouvelle fonction d'analyse de classe de stockage Amazon S3 observe les modèles d'accès aux données afin de vous aider à déterminer le moment approprié pour passer du mode `STANDARD` au mode `STANDARD_IA` (« IA » correspondant à « infrequent access », soit accès peu fréquents). Une fois que l'analyse de classe de stockage a observé les modèles d'accès peu fréquents d'un ensemble filtré de données sur une période donnée, vous pouvez utiliser les résultats d'analyse pour vous aider à améliorer vos configurations de cycle de vie. Cette fonction comprend également une analyse détaillée quotidienne de votre utilisation du stockage au niveau du compartiment, du préfixe ou de la balise que vous pouvez exporter vers un compartiment S3.	29 novembre 2016
Nouvelles récupérations de données rapides et en bloc lors de la restauration d'objets archivés de S3 Glacier.	Amazon S3 prend désormais en charge les récupérations de données rapides et en bloc en plus des récupérations standard lors de la restauration d'objets archivés sur S3 Glacier. Pour plus d'informations, consultez Restauration d'un objet archivé.	21 novembre 2016
CloudTrail journalisation des objets	CloudTrail prend en charge la journalisation des opérations d'API au niveau des objets Amazon S3`GetObject`, telles que`PutObject`, et`DeleteObject`. Vous pouvez configurer votre sélecteur d'événements de sorte qu'il enregistre les opérations API au niveau de l'objet. Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.	21 novembre 2016
US East (Ohio) Region	Amazon S3 est désormais disponible dans la Région USA Est (Ohio). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	17 octobre 2016

Modification	Description	Date
Prise en charge d'IPv6 pour Amazon S3 Transfer Acceleration	Amazon S3 prend désormais en charge le protocole Internet version 6 (IPv6) pour Amazon S3 Transfer Acceleration. Vous pouvez vous connecter à Amazon S3 via IPv6 en utilisant la nouvelle double pile pour le point de terminaison Transfer Acceleration. Pour plus d'informations, consultez Mise en route d'Amazon S3 Transfer Acceleration.	6 octobre 2016
Prise en charge d'IPv6	Amazon S3 prend désormais en charge le protocole Internet version 6 (IPv6). Vous pouvez accéder à Amazon S3 via IPv6 à l'aide de points de terminaison Dual-Stack (double pile). Pour plus d'informations, consultez Envoi de demandes à Amazon S3 via IPv6.	11 août 2016
Asia Pacific (Mumbai) Region	Amazon S3 est désormais disponible dans la Région Asie-Pacifique (Mumbai). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	27 juin 2016
Amazon S3 Transfer Acc ion	Amazon S3 Transfer Acceleration permet un transfert rapide, facile et sécurisé de fichiers sur des longues distances entre votre client et un compartiment S3. Transfer Acceleration tire parti des emplacements périphériques distribués par Amazon CloudFront dans le monde entier. Pour plus d'informations, consultez Configuration de transferts de fichiers rapides et sécurisés à l'aide d'Amazon S3 Transfer Acceleration.	19 avril 2016

Modification	Description	Date
Prise en charge du cycle de vie pour supprimer les marqueurs de suppression des objets expirés	L'action `Expiration` de configuration de cycle de vie vous permet d'indiquer à Amazon S3 de supprimer les marqueurs de suppression des objets expirés dans un compartiment avec gestion des versions. Pour plus d'informations, consultez Éléments pour décrire les actions du cycle de vie.	16 mars 2016

Modification	Description	Date
La configuration de cycle de vie du compartiment prend désormais en charge l'arrêt des chargements partitionnés inachevés	La configuration de cycle de vie du compartiment prend désormais en charge l'action `AbortIncompleteMultipartUpload` permettant de demander à Amazon S3 d'arrêter les chargements partitionnés qui ne sont pas terminés au bout d'un certain nombre de jours après leur lancement. Si un chargement partitionné est susceptible de faire l'objet d'un arrêt, Amazon S3 supprime les parties téléchargées et arrête le chargement partitionné. Pour obtenir des informations conceptuelles, consultez les rubriques suivantes dans le Guide de l'utilisateur Amazon S3 : • Interruption d'un chargement partitionné • Éléments pour décrire les actions du cycle de vie Les opérations API suivantes ont été mises à jour afin de prendre en charge cette nouvelle action : • PutBucketLifecycle – La configuration XML permet désormais de spécifier l'action `AbortIncompleteMultipartUpload` dans une règle de configuration de cycle de vie. • ListParts et InitiateMultipartUpload – Ces deux opérations API renvoient désormais deux en-têtes de réponse supplémentaires (`x-amz-abort-date` et `x-amz-abort-rule-id`) si le compartiment dispose d'une règle de cycle de vie spécifiant l'action `AbortIncompleteMultipartUpload`. Ces en-têtes dans la réponse indiquent quand le chargement partitionné peut faire l'objet d'une opération d'arrêt et quelle règle de cycle de vie s'applique.	16 mars 2016

Modification	Description	Date
Asia Pacific (Seoul) Region	Amazon S3 est désormais disponible dans la Région Asie-Pacifique (Séoul). Pour plus d'informations sur les régions et les points de terminaison Amazon S3, consultez Régions et points de terminaison dans la Références générales AWS.	6 janvier 2016
Nouvelle clé de condition et modification du chargement partitionné	Les stratégies IAM prennent désormais en charge une clé de condition `s3:x-amz-storage-class` Amazon S3. Pour plus d'informations, consultez Exemples de politiques relatives aux compartiments utilisant des clés de condition. Il n'est plus nécessaire que vous soyez à l'origine du chargement partitionné pour charger des parties et terminer le chargement. Pour plus d'informations, consultez API de chargement partitionné et autorisations.	14 décembre 2015
Région USA Standard renommée	Modification de l'intitulé de la Région « USA Standard » par « USA Est (Virginie du Nord) » Seul le nom a été mis à jour, la fonctionnalité demeure inchangée.	11 décembre 2015

Modification	Description	Date
Nouvelle classe de stockage	Amazon S3 propose désormais une nouvelle classe de stockage, STANDARD_IA (où « IA » signifie « infrequent access », ou accès peu fréquent) pour le stockage d'objets. Cette classe de stockage est optimisée pour les données à longue durée de vie et moins fréquemment consultées. Pour plus d'informations, consultez Utilisation des classes de stockage Simple Storage Service (Amazon S3). Les mises à jour apportées à la fonctionnalité de configuration de cycle de vie permettent désormais la transition d'objets vers la classe de stockage STANDARD_IA. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage. Jusqu'ici, la fonctionnalité de réplication entre Régions utilisait la classe de stockage de l'objet source pour créer un réplica de l'objet. Désormais, lorsque vous configurez la réplication entre Régions, vous pouvez spécifier une classe de stockage pour le réplica de l'objet créé dans le compartiment de destination. Pour plus d'informations, consultez Vue d'ensemble de la réplication d'objets.	16 septembre 2015
AWS CloudTrail intégration	La nouvelle AWS CloudTrail intégration vous permet d'enregistrer l'activité de l'API Amazon S3 dans votre compartiment S3. Vous pouvez l'utiliser CloudTrail pour suivre les créations ou les suppressions de compartiments S3, les modifications du contrôle d'accès ou les modifications de configuration du cycle de vie. Pour plus d'informations, consultez Journalisation des appels d'API Amazon S3 à l'aide AWS CloudTrail.	1er septembre 201

Modification	Description	Date
Augmentation de limite de compartiment	Amazon S3 prend désormais en charge les augmentations de limite de compartiment. Par défaut, les clients peuvent créer jusqu'à 100 compartiments dans leur. Compte AWS Les clients ayant besoin de compartiments supplémentaires peuvent envoyer une demande afin d'augmenter la limite du service. Pour obtenir des informations sur la manière d'augmenter votre limite de compartiment, consultez Quotas de Service AWS dans la Référence générale d'AWS . Pour plus d'informations, consultez Utilisation des AWS SDK et Limites et restrictions applicables aux compartiments.	4 août 2015
Mise à jour du modèle de cohérence (consistency)	Amazon S3 prend désormais en charge read-after-write la cohérence pour les nouveaux objets ajoutés à Amazon S3 dans la région de l'est des États-Unis (Virginie du Nord). Avant cette mise à jour, toutes les régions, à l'exception de la région USA Est (Virginie du Nord), read-after-write prenaient en charge la cohérence pour les nouveaux objets chargés sur Amazon S3. Grâce à cette amélioration, Amazon S3 assure désormais read-after-write la cohérence dans toutes les régions pour les nouveaux objets ajoutés à Amazon S3. ead-after-writeLa cohérence R vous permet de récupérer des objets immédiatement après leur création dans Amazon S3. Pour plus d'informations, consultez Régions.	4 août 2015
Notifications d'événements	La fonctionnalité de notification d'événement d'Amazon S3 a été mise à jour, ce qui vous permet désormais d'envoyer des notifications lorsque des objets sont supprimés et de filtrer les objets par nom d'objet dont le préfixe et le suffixe correspondent. Pour plus d'informations, consultez Notifications d'événements Amazon S3.	28 juillet 2015

Modification	Description	Date
CloudWatch Intégration avec Amazon	La nouvelle CloudWatch intégration d'Amazon vous permet de surveiller et de définir des alarmes relatives à votre utilisation d'Amazon S3 grâce à CloudWatch des métriques pour Amazon S3. Les indicateurs pris en charge sont la quantité totale d'octets pour le stockage standard, la quantité totale d'octets pour le stockage RRS et le nombre total d'objets pour un compartiment S3 donné. Pour plus d'informations, consultez Surveillance des métriques avec Amazon CloudWatch.	28 juillet 2015
Prise en charge de la suppression et du vidage des compartiments non vides	Amazon S3 vous permet désormais de supprimer et de vider les compartiments non vides. Pour plus d'informations, consultez Vider un compartiment.	16 juillet 2015
Stratégies de compartiment pour les points de terminaison d'un VPC Amazon	Amazon S3 a ajouté la prise en charge des stratégies de compartiment pour les points de terminaison de VPC. Vous pouvez utiliser des stratégies de compartiment S3 pour contrôler l'accès aux compartiments à partir de points de terminaison d'un VPC spécifiques ou de VPC spécifiques. Les points de terminaison d'un VPC sont faciles à configurer et assurent une connectivité fiable à Amazon S3 sans qu'une passerelle Internet ou une instance NAT soit nécessaire. Pour plus d'informations, consultez Contrôle de l'accès à partir des points de terminaison d'un VPC avec des stratégies de compartiment.	29 avril 2015
Notifications d'événements	Les notifications d'événements Amazon S3 ont été mises à jour pour permettre le passage à des autorisations basées sur les ressources pour AWS Lambda les fonctions. Pour plus d'informations, consultez Notifications d'événements Amazon S3.	9 avril 2015

Modification	Description	Date
Réplication entre Régions	Amazon S3 prend désormais en charge la réplication entre Régions. La réplication entre régions est la copie automatiq ue et asynchrone d'objets dans différents compartiments. Régions AWS Pour plus d'informations, consultez Vue d'ensemble de la réplication d'objets.	24 mars 2015
Notifications d'événements	Amazon S3 prend désormais en charge de nouveaux types d'événements et de nouvelles destinations pour la configuration des notifications de compartiment. Avant cette version, Amazon S3 ne prenait en charge que le type d'ReducedRedundancyLostObjectévénement s3 : et une rubrique Amazon SNS comme destination. Pour plus d'informations sur les nouveaux types d'événements, consultez Notifications d'événements Amazon S3.	13 novembre 2014
Chiffrement côté serveur à l'aide des clés de chiffrement fournies par le client	Chiffrement côté serveur avec clés AWS Key Management Service (AWS KMS) (SSE-KMS) Amazon S3 prend désormais en charge le chiffrement côté serveur à l'aide de. AWS KMS Cette fonctionnalité vous permet de gérer la clé d'enveloppe et d' AWS KMS appeler Amazon S3 AWS KMS pour accéder à la clé d'enveloppe dans les limites des autorisations que vous avez définies. Pour plus d'informations sur le chiffrement côté serveur avec AWS KMS, voir Protection des données à l'aide du chiffrement côté serveur avec. AWS Key Management Service	12 novembre 2014
Europe (Frankfurt) Region	Amazon S3 est maintenant disponible dans la Région Europe (Francfort).	23 octobre 2014

Modification	Description	Date
Chiffrement côté serveur à l'aide des clés de chiffrement fournies par le client	Amazon S3 prend désormais en charge le chiffrement côté serveur avec les clés de chiffrement fournies par le client (SSE-C). Le chiffrement côté serveur vous permet de demander à Amazon S3 de chiffrer vos données inactives . Lorsque vous utilisez SSE-C, Amazon S3 chiffre vos objets grâce aux clés de chiffrement personnalisées que vous fournissez. Étant donné qu'Amazon S3 effectue le chiffrement pour vous, vous bénéficiez de l'avantage d'utiliser vos propres clés de chiffrement sans encourir le coût lié à l'écriture ou l'exécution de votre propre code de chiffrement. Pour plus d'informations sur SSE-C, consultez Chiffrement côté serveur (à l'aide des clés de chiffrement fournies par le client).	12 juin 2014
Prise en charge du cycle de vie pour la gestion des versions	Avant cette publication, la configuration de cycle de vie était prise en charge uniquement sur les compartiments non versionnés. Vous pouvez désormais configurer le cycle de vie aussi bien sur les compartiments non versionné s que les compartiments pour lesquels la gestion des versions d'objet est activée. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.	20 mai 2014
Révision des rubriques consacrées au contrôle d'accès	Révision de la documentation consacrée au contrôle d'accès Amazon S3. Pour plus d'informations, consultez Identity and Access Management pour Amazon S3.	15 avril 2014
Révision de la rubrique consacrée à la journalisation des accès au serveur	Révision de la documentation consacrée à la journalisation des accès au serveur. Pour plus d'informations, consultez Enregistrement de demandes avec journalisation des accès au serveur.	26 novembre 2013
Exemples SDK pour .NET mis à jour dans la version 2.0	Les exemples SDK pour .NET fournis dans ce guide sont désormais compatibles avec la version 2.0.	26 novembre 2013

Modification	Description	Date
Obsolescence de la prise en charge SOAP via HTTP	La prise en charge de SOAP via HTTP est obsolète, mais continue d'être disponible sur HTTP. Les nouvelles fonctionnalités Amazon S3 ne sont pas prises en charge pour SOAP. Nous vous recommandons d'utiliser l'API REST ou les AWS SDK.	20 septembre 201
Prise en charge des variables dans la stratégie IAM	Le langage de la politique IAM prend désormais en charge des variables. Lorsqu'une stratégie est évaluée, toutes les variables de stratégie sont remplacées par des valeurs issues d'informations basées sur le contexte provenant de la session de l'utilisateur authentifié. Vous pouvez utiliser des variables de stratégie pour définir des stratégie s à usage général sans dresser de manière explicite la liste des éléments composant cette stratégie. Pour plus d'informations sur les variables de stratégie, consultez Présentation des variables de la stratégie IAM dans le Guide de l'utilisateur IAM. Pour obtenir des exemples de variables de stratégies dans Amazon S3, veuillez consulter Exemples de politiques basées sur l'identité pour Amazon S3.	3 avril 2013
Prise en charge via la console du paiement par le demandeur	Vous pouvez désormais configurer votre compartiment pour le Paiement par le demandeur via la console Amazon S3. Pour plus d'informations, consultez Utilisation de compartiments de paiement par le demandeur pour les transferts de stockage et l'utilisation.	31 décembre 2012

Modification	Description	Date
Prise en charge du domaine racine pour l'hébergement de site Web	Amazon S3 prend désormais en charge l'hébergement de sites web dans le domaine racine. Les internautes de votre site Internet peuvent ainsi accéder à votre site à partir de leur navigateur sans devoir saisir www dans l'adresse Web (par exemple, ils peuvent utiliser example.com au lieu de www.example.com). De nombreux clients hébergent déjà des sites web statiques sur Amazon S3, accessibles à partir d'un sous-domaine www (par exemple, www.example.com). Jusqu'ici, pour permettre l'accès au domaine racine, vous deviez exécuter votre propre serveur web afin qu'il fasse office de proxy pour les demandes du domaine racine depuis les navigateurs jusqu'à votre site web Amazon S3. L'exécution d'un serveur Web vers des demandes proxy entraînait des frais supplémentaires, alourdissait la charge opérationnelle et introduisait un point de défaillance potentiel. Vous pouvez désormais profiter de la grande disponibilité et durabilité d'Amazon S3 aussi bien pour les adresses de type www que les adresses de domaine racine. Pour plus d'informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.	27 décembre 2012
Révision de la console	Nous avons mis à jour la console Amazon S3. Les rubriques de la documentation se rapportant à la console ont donc été révisées en conséquence.	14 décembre 2012

Modification	Description	Date
Prise en charge de l'archivage des données dans S3 Glacier	Amazon S3 prend désormais en charge une option de stockage qui vous permet d'exploiter le service de stockage économique de S3 Glacier pour l'archivage des données. Pour archiver des objets, vous devez définir des règles d'archivage identifiant des objets ainsi qu'un calendrier pour indiquer quand Amazon S3 doit archiver ces objets sur S3 Glacier. Vous pouvez facilement définir les règles d'un compartiment à l'aide de la console Amazon S3 ou par programmation à l'aide de l'API ou AWS des kits SDK Amazon S3. Pour plus d'informations, consultez Gestion du cycle de vie de votre stockage.	13 novembre 2012
Prise en charge des redirections de page de site Web	Pour un compartiment configuré comme un site web, Amazon S3 prend désormais en charge la redirection d'une demande pour un objet vers un autre objet dans le même compartiment ou vers une URL externe. Pour plus d'informations, consultez (Facultatif) Configuration de la redirection de pages web. Pour obtenir des informations sur l'hébergement de sites Web, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.	4 octobre 2012

Modification	Description	Date
Prise en charge du partage des ressources cross-origin (CORS)	Amazon S3 prend désormais en charge le partage des ressources cross-origin (CORS). Le mécanisme CORS permet aux applications Web clientes chargées dans un domaine donné d'interagir avec les ressources d'un autre domaine ou d'y accéder. Avec le support CORS dans Amazon S3, vous pouvez créer des applications web clientes denses riches sur Amazon S3 et de manière sélective permettre un accès cross-domaine à vos ressources Amazon S3. Pour plus d'informations, consultez Utilisation du partage des ressources entre origines multiples (CORS).	31 août 2012
Prise en charge des balises de répartition des coûts	Amazon S3 prend désormais en charge le balisage de répartition des coûts qui permet d'étiqueter les compartiments S3 et ainsi de faciliter le suivi de leurs coûts en fonction des projets ou d'autres critères. Pour plus d'informations sur le balisage des compartiments, consultez Utilisation des balises de répartition des coûts pour les compartiments S3.	21 août 2012

Modification	Description	Date
Prise en charge des stratégies de compartiment pour l'accès aux API protégé par MFA	Amazon S3 prend désormais en charge l'accès aux API protégé par MFA, une fonctionnalité qui permet d'appliquer l'authentification AWS multifactorielle pour un niveau de sécurité supplémentaire lors de l'accès à vos ressources Amazon S3. Il s'agit d'une fonctionnalité de sécurité qui exige des utilisateurs qu'ils prouvent qu'ils détiennent physiquement un appareil MFA en fournissant un code MFA valide. Pour plus d'informations, consultez Authentification multifacteur AWS. Vous pouvez désormais exiger une authentification MFA pour toutes les demandes d'accès à vos ressources Amazon S3. Pour appliquer l'authentification MFA, Amazon S3 prend désormais en charge la clé `aws:MultiFactorAuthAge` dans une stratégie de compartiment. Pour un exemple de stratégie de compartiment, consultez Exigence d'une MFA.	10 juillet 2012
Prise en charge de la fonctionnalité d'expiration d'objet	La fonctionnalité d'expiration d'objet vous permet de planifier la suppression automatique de données une fois écoulé un certain délai. Pour définir l'expiration d'objet, vous ajoutez une configuration de cycle de vie à un compartiment.	27 décembre 2011
Prise en charge d'une nouvelle Région	Amazon S3 prend désormais en charge la Région Amérique du Sud (São Paulo). Pour plus d'informations, consultez Accès à un compartiment Amazon S3 et liste des compartiments.	14 décembre 2011

Modification	Description	Date
Suppression de plusieurs objets	Amazon S3 prend désormais en charge l'API de suppression de plusieurs objets qui vous permet de supprimer plusieurs objets dans le cadre d'une seule demande. Grâce à cette fonction, vous pouvez supprimer un grand nombre d'objets d'Amazon S3 bien plus rapidement qu'avec plusieurs demandes DELETE individuelles. Pour plus d'informations, consultez Suppression d'objets Amazon S3.	7 décembre 2011
Prise en charge d'une nouvelle Région	Amazon S3 prend désormais en charge la Région USA Ouest (Oregon). Pour plus d'informations, consultez Compartiments et Régions.	8 novembre 2011
Mise à jour de la documentation	Correctifs de la documentation.	8 novembre 2011
Mise à jour de la documentation	En plus des correctifs de la documentation, cette mise à jour offre les améliorations suivantes : • Nouvelles sections de chiffrement côté serveur utilisant le AWS SDK for PHP et le AWS SDK for Ruby (voirSpécification du chiffrement côté serveur avec des clés gérées par Amazon S3 (SSE-S3)).	17 octobre 2011
Prise en charge du chiffrement côté serveur	Amazon S3 prend désormais en charge le chiffrement côté serveur. Ceci vous permet de demander à Amazon S3 de chiffrer vos données inactives, c'est-à-dire de chiffrer vos données d'objets l'objet lorsqu'Amazon S3 les écrit sur des disques dans ses centres de données. Outre les mises à jour de l'API REST, .NET AWS SDK for Java et .NET fournissent les fonctionnalités nécessaires pour demander un chiffrement côté serveur. Vous pouvez également demander le chiffrement côté serveur lors du chargement d'objet avec la AWS Management Console. Pour en savoir plus sur le chiffrement des données, veuillez consulter la section Utilisation du chiffrement des données.	4 octobre 2011

Modification	Description	Date
Mise à jour de la documentation	En plus des correctifs de la documentation, cette mise à jour offre les améliorations suivantes : • Exemples Ruby et PHP ajoutés dans la section Demandes. • Ajout de sections décrivant comment générer et utiliser des URL pré-signées. Pour plus d'informations, consultez Partage d'objets à l'aide d'URL présignées et Partage d'objets à l'aide d'URL présignées. • Mise à jour d'une section existante pour présenter AWS les explorateurs pour Eclipse et Visual Studio. Pour plus d'informations, consultez Développement avec Amazon S3 à l'aide des AWS SDK.	22 septembre 201

Modification	Description	Date
Prise en charge des demandes utilisant des informations d'identification de sécurité temporaires	En plus d'utiliser vos informations d'identification de sécurité Compte AWS et celles de l'utilisateur IAM pour envoyer des demandes authentifiées à Amazon S3, vous pouvez désormais envoyer des demandes à l'aide des informations d'identification de sécurité temporaires que vous avez obtenues auprès de AWS Identity and Access Management (IAM). Vous pouvez utiliser l' AWS Security Token Service API ou les bibliothèques d' AWS encapsulation du SDK pour demander ces informations d'identification temporaires à IAM. Vous pouvez demander ces informations d'identification de sécurité temporaires pour votre propre utilisation ou pour les passer à des utilisateurs fédérés et à des applications. Cette fonctionnalité vous permet de gérer vos utilisateurs à l'extérieur AWS et de leur fournir des informations de sécurité temporaires pour accéder à vos AWS ressources. Pour plus d’informations, consultez Demandes. Pour plus d’informations sur la prise en charge des informations d’identification de sécurité temporaires par IAM, consultez Informations d’identification de sécurité temporaires dans le Guide de l’utilisateur IAM.	3 août 2011

Modification	Description	Date
L'API de chargement partitionné peut désormais copier des objets d'une taille maximale de 5 To	Avant cette mise à jour, l'API Amazon S3 prenait en charge la copie d'objets d'une taille maximale de 5 Go. Pour activer la copie d'objets d'une taille supérieure à 5 Go, Amazon S3 enrichit désormais l'API de chargement partitionné d'une nouvelle opération , `Upload Part (Copy)` . Vous pouvez utiliser cette opération de chargement partitionné pour copier les objets d'une taille maximale de 5 To. Pour plus d'informations, consultez Copier, déplacer et renommer des objets. Pour obtenir des informations conceptuelles sur l'API de chargement partitionné, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.	21 juin 2011
Désactivation des appels de l'API SOAP via HTTP	Pour améliorer la sécurité, les appels de l'API SOAP via HTTP ont été désactivés. Les demandes SOAP authentifiées et anonymes doivent être envoyées à Amazon S3 à l'aide de SSL.	6 juin 2011
IAM permet la délégation entre comptes	Auparavant, pour accéder à une ressource Amazon S3, un utilisateur IAM avait besoin des autorisations du parent Compte AWS et du propriétaire de la ressource Amazon S3. Grâce à l'accès entre comptes, l'utilisateur IAM a besoin désormais uniquement de l'autorisation du compte propriétaire. En d'autres termes, si un propriétaire de ressource accorde l'accès à un Compte AWS, il Compte AWS peut désormais accorder à ses utilisateurs IAM l'accès à ces ressources. Pour plus d'informations, veuillez consulter Création d'un rôle pour déléguer des autorisations à un utilisateur IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la désignation de mandataires dans une stratégie de compartiment, consultez Principes relatifs aux politiques relatives aux compartiments.	6 juin 2011

Modification	Description	Date
Nouveau lien	Les informations relatives au point de terminaison de ce service sont désormais situées dans la Référence générale AWS . Pour plus d'informations, consultez Regions and Endpoints (Régions et points de terminaison) dans la Référence générale AWS.	1 mars 2011
Prise en charge de l'hébergement de sites web statiques dans Amazon S3	Amazon S3 offre désormais une prise en charge améliorée de l'hébergement de sites web statiques. Parmi ces améliorations, citons la prise en charge des documents d'index et des documents d'erreur personnalisés. Lorsque vous utilisez ces fonctions, les demandes effectuées à la racine de votre compartiment ou dans un sous-dossier (par exemple, `http://mywebsite.com/subfolder`) renvoient votre document d'index en lieu et place de la liste des objets contenus dans votre compartiment.. En cas d'erreur, Amazon S3 renvoie vos messages d'erreur personnalisés en lieu et place des messages d'erreur d'Amazon S3. Pour plus d’informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.	6 juin 2011
Les informations relatives au point de terminaison de ce service sont désormais situées dans la Référence générale AWS . Pour plus d'informations, consultez Regions and Endpoints (Régions et points de terminaison) dans la Référence générale AWS.	1 mars 2011	

Modification	Description	Date
Prise en charge de l'hébergement de sites web statiques dans Amazon S3	Amazon S3 offre désormais une prise en charge améliorée de l'hébergement de sites web statiques. Parmi ces améliorations, citons la prise en charge des documents d'index et des documents d'erreur personnalisés. Lorsque vous utilisez ces fonctions, les demandes effectuées à la racine de votre compartiment ou dans un sous-dossier (par exemple, `http://mywebsite.com/subfolder`) renvoient votre document d'index en lieu et place de la liste des objets contenus dans votre compartiment.. En cas d'erreur, Amazon S3 renvoie vos messages d'erreur personnalisés en lieu et place des messages d'erreur d'Amazon S3. Pour plus d'informations, consultez Hébergement d'un site Web statique à l'aide d'Amazon S3.	17 février 2011
Prise en charge par l'API des en-têtes de réponse	L'API REST de GET Object vous permet désormais de modifier les en-têtes de réponse de la demande GET Object de REST, pour chaque demande. Ainsi, vous pouvez modifier des métadonnées d'objet dans la réponse, sans altérer l'objet lui-même. Pour plus d'informations, consultez Téléchargement d'objets.	14 janvier 2011
Prise en charge des objets volumineux	Amazon S3 a augmenté la taille maximale d'un objet à stocker dans un compartiment S3 : celle-ci passe de 5 Go à 5 To. Si vous utilisez l'API REST, vous pouvez charger des objets de 5 Go maximum en une seule opération PUT. Pour les objets plus volumineux, vous devrez utiliser le chargement partitionné de l'API REST pour charger des objets en plusieurs parties. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.	9 décembre 2010

Modification	Description	Date
Chargement partitionné	Le chargement partitionné permet des téléchargements plus rapides et plus flexibles sur Amazon S3. Il vous permet de charger un seul objet en tant qu'ensemble de parties. Pour plus d'informations, consultez Chargement et copie d'objets à l'aide d'un chargement partitionné.	10 novembre 2010
Prise en charge de l'ID canonique dans les stratégies de compartiment	Vous pouvez désormais spécifier des ID canoniques dans les stratégies de compartiment. Pour plus d'informations, consultez Principes relatifs aux politiques relatives aux compartiments.	17 septembre 2010
Amazon S3 fonctionne avec IAM	Ce service s'intègre désormais à AWS Identity and Access Management (IAM). Pour plus d'informations, consultez Services Services AWS qui fonctionnent avec IAM dans le Guide de l'utilisateur IAM.	2 septembre 2010
Notifications	La fonction de notifications Amazon S3 vous permet de configurer un compartiment afin qu'Amazon S3 publie un message vers une rubrique Amazon Simple Notification Service (Amazon SNS) lorsqu'Amazon S3 détecte un événement clé sur un compartiment. Pour plus d'informations, consultez Configuration de la notification des événements de compartiment.	14 juillet 2010
Stratégies de compartiment	Les stratégies de compartiment sont un système de gestion des accès permettant d'accorder des autorisations dans des compartiments, objets et groupes d'objets. Cette fonctionnalité s'ajoute, et dans de nombreux cas, remplace les listes de contrôle d'accès. Pour plus d'informations, consultez Politiques relatives aux compartiments pour Amazon S3.	6 juillet 2010

Modification	Description	Date
Syntaxe de type chemin disponible dans toutes les Régions	Amazon S3 prend désormais en charge la syntaxe de type chemin pour tout compartiment dans une Région USA classique, ou si le compartiment se trouve dans la même Région, au point de terminaison de la demande. Pour plus d'informations, consultez Hébergement virtuel.	9 juin 2010
Nouveau point de terminaison pour l'Europe (Irlande)	Simple Storage Service (Amazon S3) fournit désormais un point de terminaison pour l'Europe (Irlande) : `http://s3-eu-west-1.amazonaws.com` .	9 juin 2010
Console	Vous pouvez maintenant utiliser Simple Storage Service (Amazon S3) via la AWS Management Console. Vous pouvez lire toutes les fonctions d'Amazon S3 dans la console dans le Guide de l'utilisateur Amazon Simple Storage Service.	9 juin 2010
Redondance réduite	Amazon S3 vous permet désormais de réduire vos coûts de stockage en stockant des objets dans Amazon S3, avec moins de redondance. Pour plus d'informations, consultez Stockage à redondance réduite (RRS).	12 mai 2010
Prise en charge d'une nouvelle Région	Amazon S3 prend désormais en charge la Région Asie-Pacifique (Singapour). Pour plus d'informations, consultez Compartiments et Régions.	28 avril 2010
Contrôle de version d'un objet	Cette mise à jour présente la gestion des versions d'objets. Tous les objets peuvent désormais avoir une clé et une version. Lorsque vous activez la gestion des versions pour un compartiment, Amazon S3 confère un ID de version unique à tous les objets qui y sont ajoutés. Cette fonctionnalité permet la récupération en cas de suppressions ou de remplacements involontaires. Pour plus d'informations, consultez Gestion des versions et Utilisation de la gestion des versions.	8 février 2010

Modification	Description	Date
Prise en charge d'une nouvelle Région	Amazon S3 prend désormais en charge la Région USA Ouest (Californie du Nord). Le nouveau point de terminaison pour les demandes relatives à cette région est `s3-us-west-1.amazonaws.com` . Pour plus d'informations, consultez Compartiments et Régions.	2 décembre 2009
AWS SDK for .NET	AWS fournit désormais des bibliothèques, des exemples de code, des didacticiels et d'autres ressources aux développeurs de logiciels qui préfèrent créer des applications à l'aide d'opérations d'API spécifiques au langage .NET plutôt que REST ou SOAP. Ces bibliothèques offrent des fonctions de base (non présentes dans les API REST ou SOAP), telles que l'authentification de demande, les nouvelles tentatives de demande et la gestion des erreurs ; celles-ci vous permettent de démarrer plus facilement. Pour plus d'informations sur les librairies et ressources de langage spécifique, consultez Développement avec Amazon S3 à l'aide des AWS SDK.	11 novembre 2009

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le Glossaire AWS dans la Référence Glossaire AWS.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.